MVUS 8085

Systembeschreibung

Schaltpläne und Signalpläne der Bausteine

Bauteilliste und Bestückungsplan

Leiterbahnführungen der Platine (Layout)

Bedienungsanleitung des Monitors

Monitorlisting

MVUS 8085 Systembeschreibung

Das Mikrocomputer Versuchs- und Übungsssystem MVUS 8085 dient zum Testen von Versuchs- und Übungsprogrammen für den Mikroprozessor 8085. Bild 1 zeigt eine Übersicht. Das System enthält:

- einen Mikroprozessor 8085 mit einem 4-MHz-Quarz. Taktfrequenz 2 MHz (Zykluszeit 500 ns).
- einen EPROM 2716 (2 K Byte) auf den Adressen 0000 bis 07FF mit einem Monitor für hexadezimale Eingabe und Anzeige.
- Zwei 24polige Sockel wahlweise für den EPROM-Typ 2716 oder den austauschbaren RAM-Typ 6116 oder 2016 auf den Adressen 1000 bis 17FF und 1800 bis 1FFF. Über zwei Brücken lassen sich die RAM-Bausteine gegen Beschreiben schützen.
- Einen Mehrzweckbaustein 8155 (RAM/IO/TIMER) für den Betrieb einer 9stelligen Siebensegmentanzeige sowie eines Tastenfeldes von 16 Zifferntasten und 8 Kommandotasten für die Eingabe von hexadezimalen Adressen und Daten sowie Kommandos.
- Eine Prozessorsteuerung für RESET, Programmunterbrechung (TRAP) und eine Einzeltaktsteuerung, mit der Befehle in einzelnen Operationszyklen ausgeführt werden können (READY).
- Eine Parallelschnittstelle 8255, mit der wahlweise über einen 24poligen Programmiersockel EPROMs vom Typ 2716 programmiert werden können oder an die ein Ein/Ausgabefeld mit Kippschaltern und Leuchtdioden angeschlossen werden kann.
- Eine vierstellige hexadezimale Adreßbusanzeige und eine zweistellige hexadezimale Datenbusanzeige, die wahlweise auch als Ausgabeport (Adresse 18H) verwendet werden kann.
- Acht Schiebeschalter, die wahlweise als Eingabeport (Adresse 10H) oder als Eingabe für das INTA-Signal verwendet werden können.
- Eine 31polige Steckverbinder-Buchse mit allen Bussignalen für Systemerweiterungen oder den Anschluß eines Steckpultes für Versuchsschaltungen oder für Messungen vom Bussignalen.
- Eine serielle Schnittstelle (V24) für den Anschluß an ein Terminal oder Host-System (Alphatronic). Der Monitor enthält Treiberprogramme für die serielle Datenübertragung mit SID und SOD.

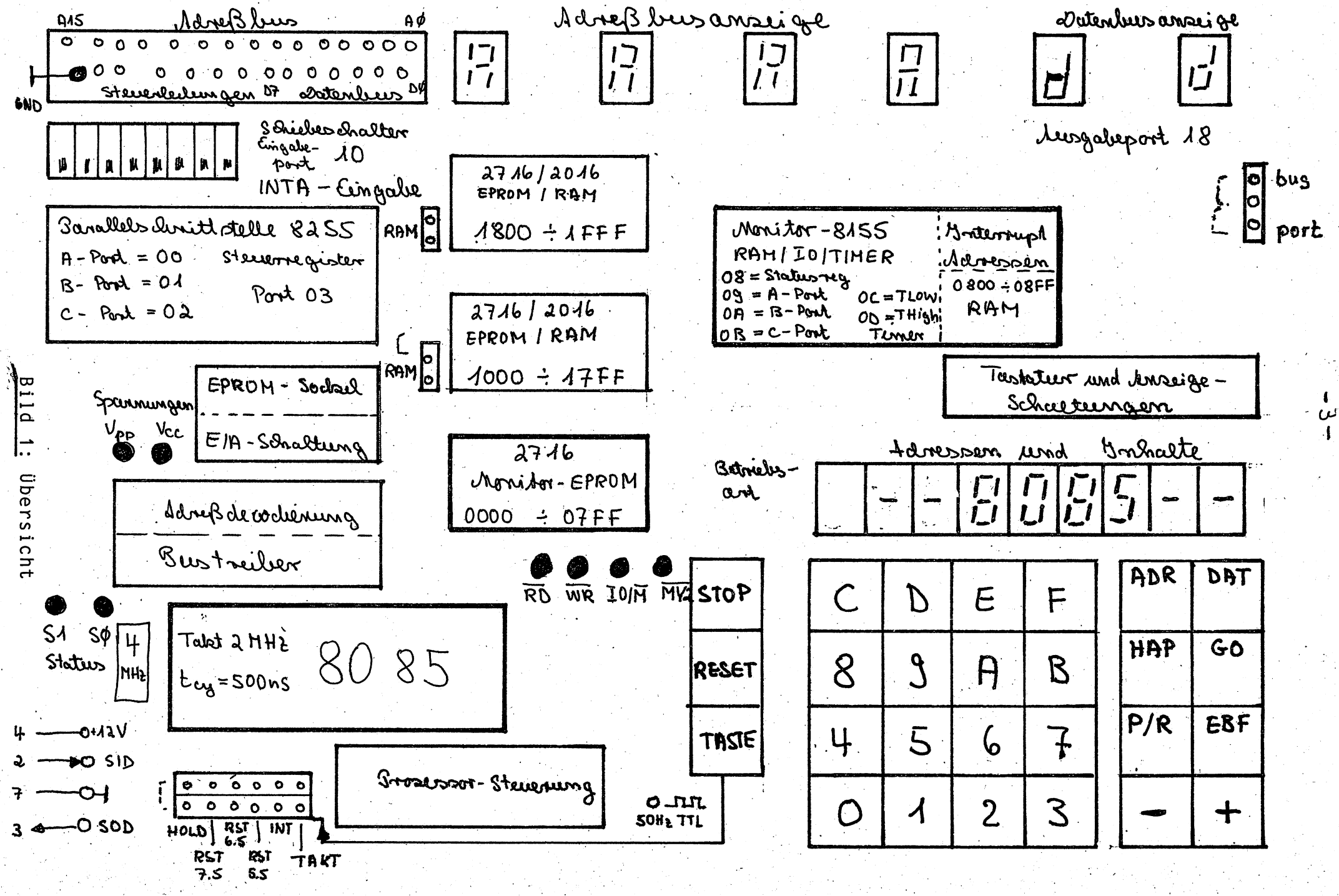


Bild 1: Übersicht

Prozessorsteuerung

Bild 2 zeigt den Schaltplan. Der RESIN-Eingang des Prozessors wird mit der entprellten Taste RES betätigt. Beim Einschalten der Versorgungsspannung sorgt ein RC-Glied für ein automatisches RESET, mit dem der Monitor gestartet wird.

Der TRAP-Interrupt kann wahlweise über die STOP-Taste (Programmabbruch) oder über einen Timer ausgelöst werden (Einzelschritt). Die TRAP-Interruptadresse wird durch den Monitor auf die RAM-Adresse 0824H umgelenkt, die jedoch bei jedem Start eines Benutzerprogramms mit einer Monitoradresse belegt wird und vom Benutzer nur durch ein laufendes Programm geändert werden kann.

Die dritte entprellte TASTE kann über ein Brückenfeld wahlweise sechs verschiedene Funktionen auslösen:

- Die HOLD-Brücke (links) bringt den Prozessor in den HALT-Zustand, mit dem alle Bussignale tristate werden.
- Die RST-7.5-Brücke löst einen RST-7.5-Interrupt aus. Die Startadresse des Interruptprogramms wird durch den Monitor auf die RAM-Adresse 083CH umgeleitet.
- Die RST-6.5-Brücke löst einen RST-6.5-Interrupt aus. Die Startadresse des Interruptprogramms wird durch den Monitor auf die RAM-Adresse 0834H umgeleitet.
- Die RST-5.5-Brücke löst einen RST-5.5-Interrupt aus. Die Startadresse des Interruptprogramms wird durch den Monitor auf die RAM-Adresse 082CH umgeleitet.
- Die INT-Brücke löst einen INTR-Interrupt aus. Das INTA-Signal holt von den Schiebeschaltern den Funktionscode eines Befehls. Der RST-0-Befehl startet den Monitor. Der RST-7-Befehl ist für Haltepunkte belegt. Alle anderen RST-Befehle stehen über RAM-Adressen zur Verfügung.
- Die TAKT-Brücke dient zur Ausführung eines Programms im Einzeltakt. Dabei wird der Prozessor über den READY-Eingang in jedem Maschinenzyklus angehalten. Alle Bus- und Steuersignale bleiben dabei erhalten und können an der Bus- und Steuersignalananzeige verfolgt werden.

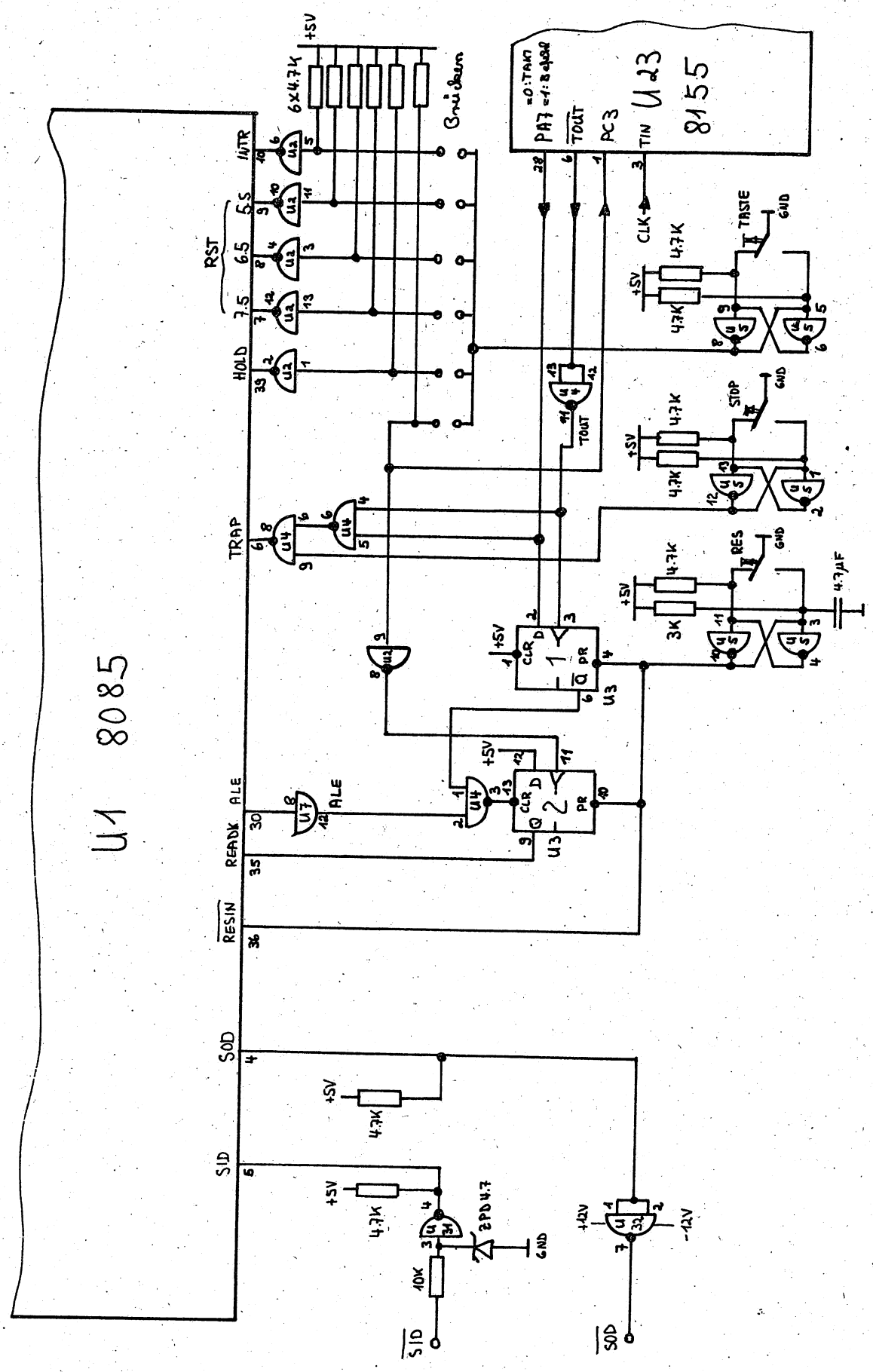


Bild 2: Schaltplan Prozessorsteuerung

Bustreiber und Adreßdecodierung

Bild 3 zeigt den Schaltplan. Die Steuersignale CLK, HLDA, S1 und S0 werden immer verstärkt. Aus den Signalen HLDA, S1 und S0 wird ein Signal TC gebildet, mit dem die übrigen Treiber in den tristate-Zustand gebracht werden können. Die Bedingung lautet: HOLD-Anforderung oder S1 UND S0 = LOW (HALT-Befehl).

Die Adreßdecodierung wird getrennt nach Speicherbereich und Ein/Ausgabebereich. Alle Bausteine werden nur für A15=A14=A13=LOW ausgewählt. Die restlichen sieben Auswahlbedingungen sind frei für Systemerweiterungen.

Speicherbereich des Systems

		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Monitor EPROM	0000 07FF	0 0	0 0	0 0	0 0	0 0	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1
Monitor 8155 RAM	0800 08FF	0 0	0 0	0 0	0 0	1 1	X X	X X	X X	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1
Benutzer RAM I	1000 17FF	0 0	0 0	0 0	1 1	0 0	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1
Benutzer RAM II	1800 1FFF	0 0	0 0	0 0	1 1	1 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1

Ein/Ausgabebereich des Systems

		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8255 E/A-Ports	00 03	0 0	0 0	0 0	0 0	0 0	X X	0 1	0 1	0 0	0 0	0 0	0 0	0 0	X X	0 1	0 1
8155 (Moni) E/A-Ports	08 0F	0 0	0 0	0 0	0 0	1 1	0 1	0 1	0 1	0 0	0 0	0 0	0 0	1 1	0 1	0 1	0 1
Schiebe-Schalter	10	0	0	0	1	0	X	X	X	0	0	0	1	0	X	X	X
Anzeige-Einheit	18	0	0	0	1	1	X	X	X	0	0	0	1	1	X	X	X

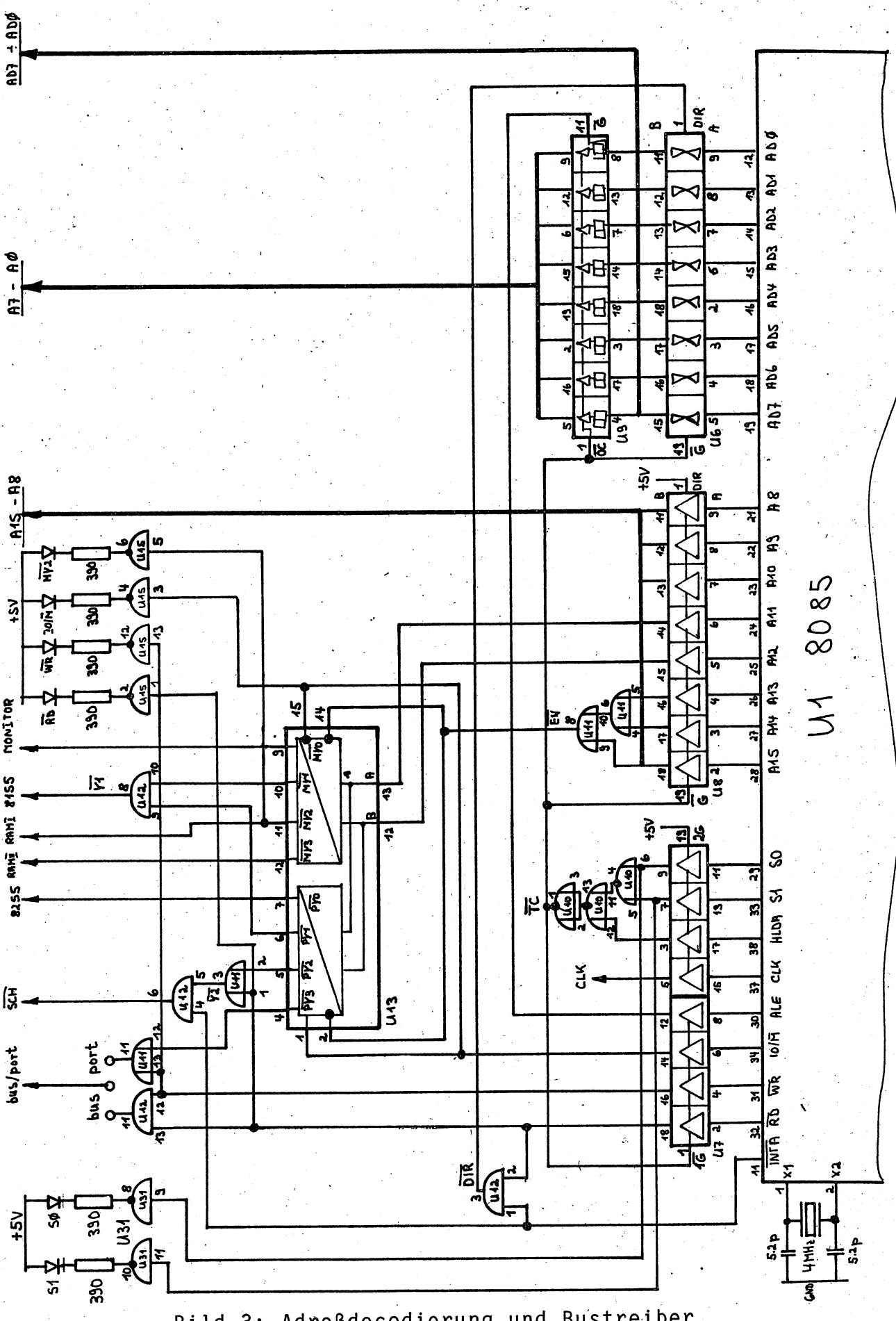


Bild 3: Adreßdecodierung und Bustreiber

Monitor-Eingabe und -Anzeige und Speicherbausteine

Bild 4 zeigt den Schaltplan. Die Anschlüsse PC0 , PC1, PC2, PA4, PA5 und PA6 des Bausteins 8155 (U23) werden für die Eingabe bzw. Anzeige nicht benötigt und sind auf sechs Pfosten geführt. Die Anzeige arbeitet im Multiplexverfahren. Die Katodensegmente aller Anzeigestellen sind miteinander verbunden. Über den 1-aus-16 -Decoder 74154 (U16) wird eine Stelle ausgewählt, und die entsprechende Anode wird an +5V gelegt. Die Eingabetastatur besteht aus 2 Zeilen und 16 Spalten. Die Zeilenleitungen werden über Widerstände auf HIGH gehalten; die Spaltenleitungen werden nacheinander über den 1-aus-16-Decoder auf LOW gelegt.

Die beiden Sockel U18 und U20 können wahlweise mit EPROMs vom Typ 2716 oder RAMs vom Typ 6116 oder 2016 bestückt werden. Sind die beiden Brücken gesteckt, so können die RAM-Bausteine beschrieben werden. Sind die Brücken offen, so ist kein Schreiben möglich. Damit lassen sich Programme in der Testphase gegen Selbstzerstörung schützen.

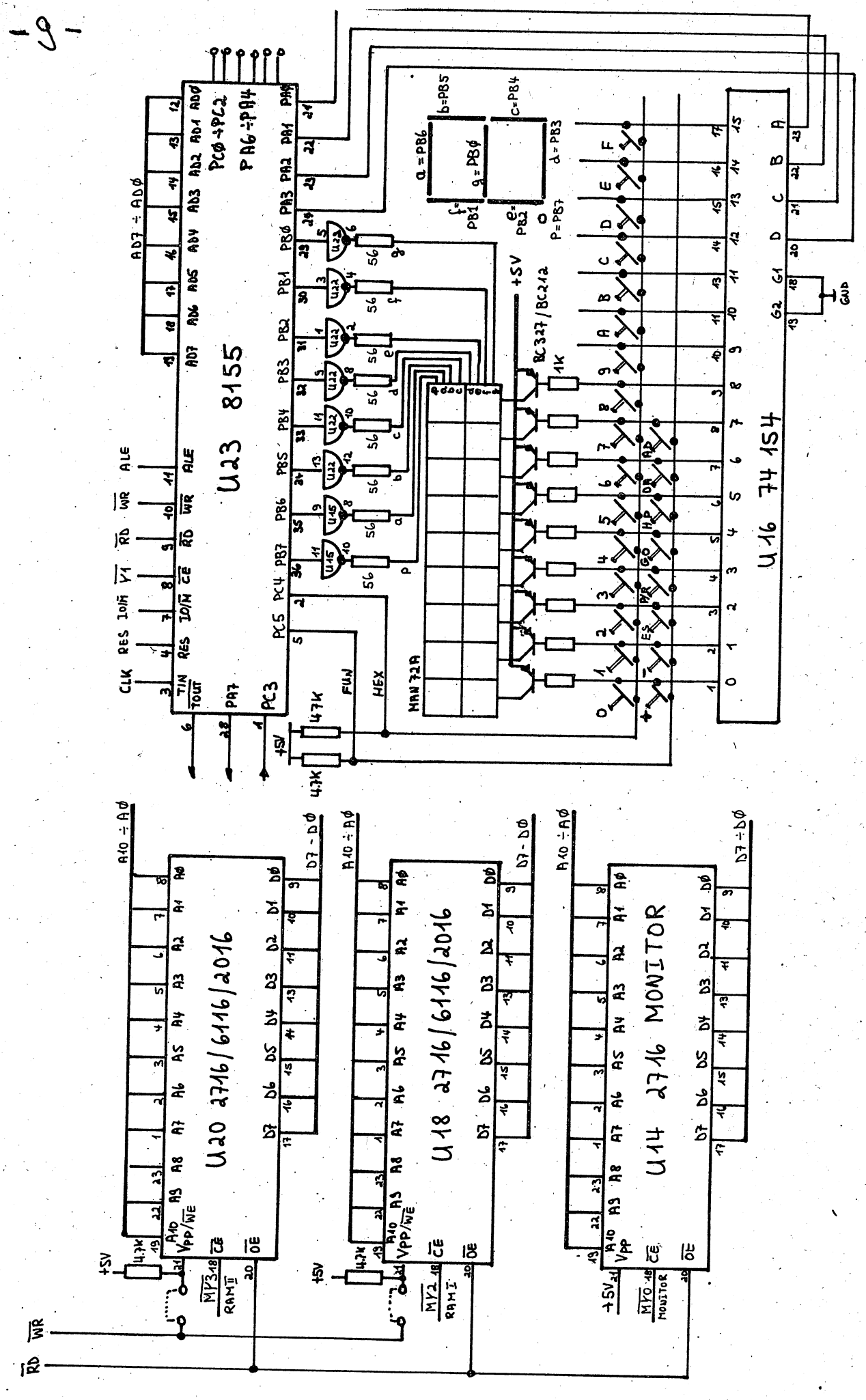


Bild 4: Eingabe und Anzeige und Speicherbausteine

Busanzeige und Programmiersockel.

Bild 5 zeigt die Schaltung. Die vier linken oberen Siebensegmentanzeigen zeigen über Hexadezimal-zu-Siebensegmentdecoder den Zustand des Adreßbus an. Die beiden rechten Anzeigen lassen sich über Brücken umstecken. Brücke oben: Busanzeige; Brücke unten: Ausgabeport 18H.

Mit den acht Schiebeschaltern in der linken oberen Ecke lassen sich Potentiale auf den Datenbus legen. Stellung oben: HIGH; Stellung unten: LOW. Die Übernahme erfolgt entweder mit dem Signal INTA nach einem INTR-Interrupt oder durch die Port-Adresse 10H

Der Programmiersockel (U17) dient wahlweise zum Programmieren von EPROMs vom Typ 2716 oder zur Verbindung mit einer Ein/Ausgabeschaltung für Kippschalter und Leuchtdioden. In diesem Fall enthält der Sockel keinen Anschluß für + 5 Volt; die Leitung Vcc wird über PC6 und einen Transistorschalter auf +5 Volt gelegt. Die Verbindung mit der Speisespannung sollte über einen Klemmkaken erfolgen. Die drei Anschlüsse PC5, PC6 und PC7 sind auf drei Pfosten geführt.

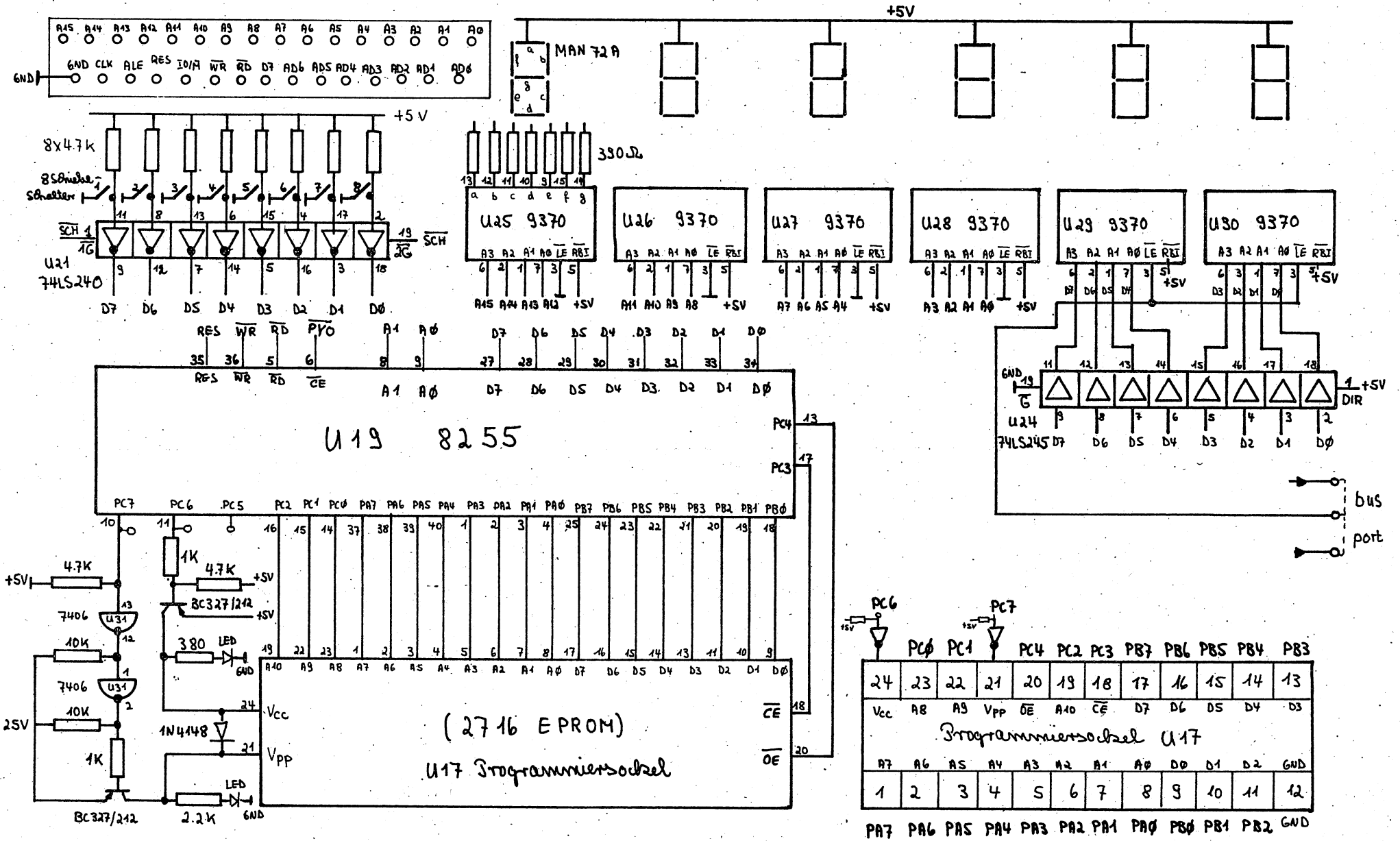


Bild 5: Busanzeige und Bus-Ein-Ausgabe

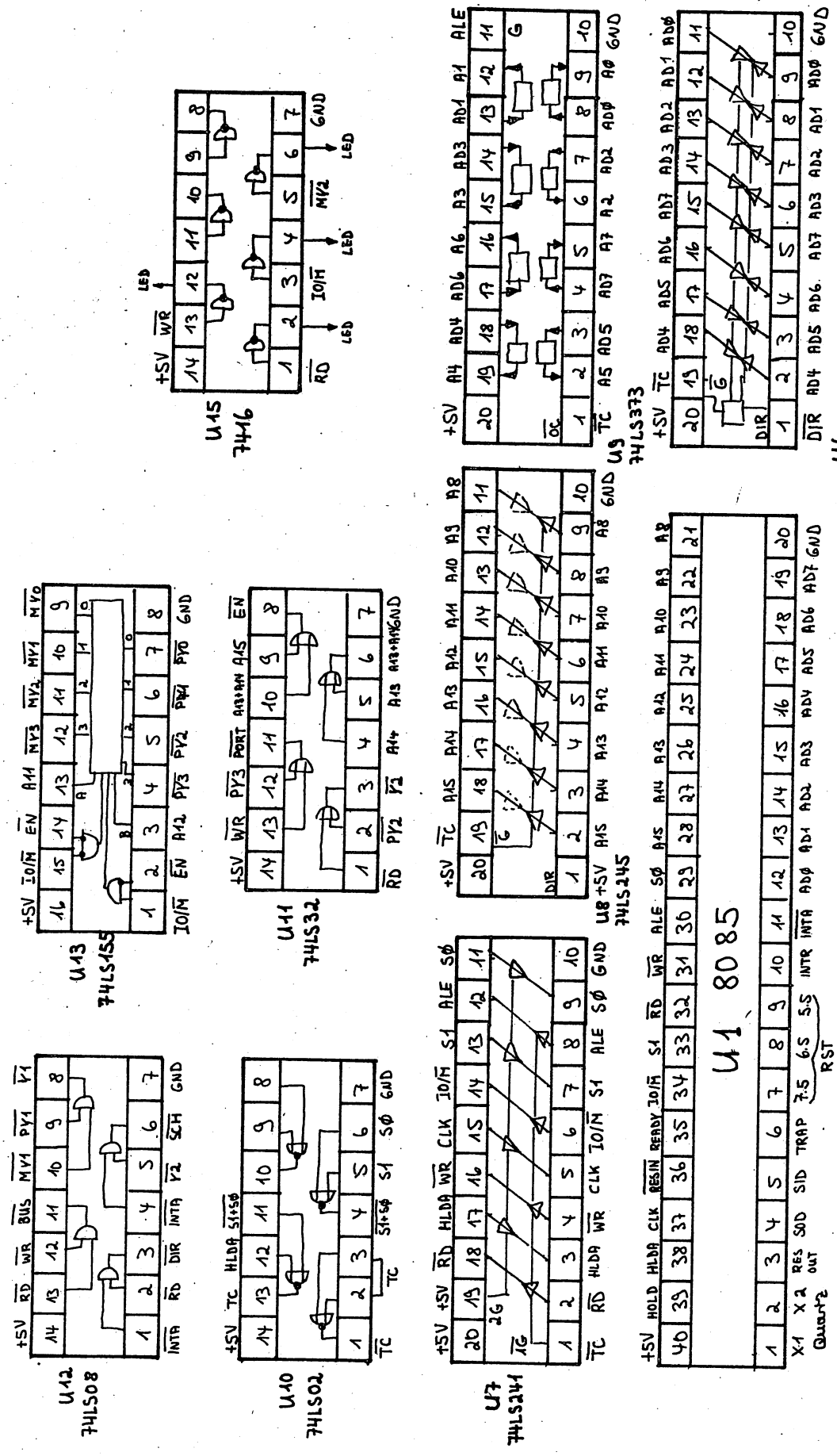


Bild 6: Signalplan Treiber und Adreßdecodierung

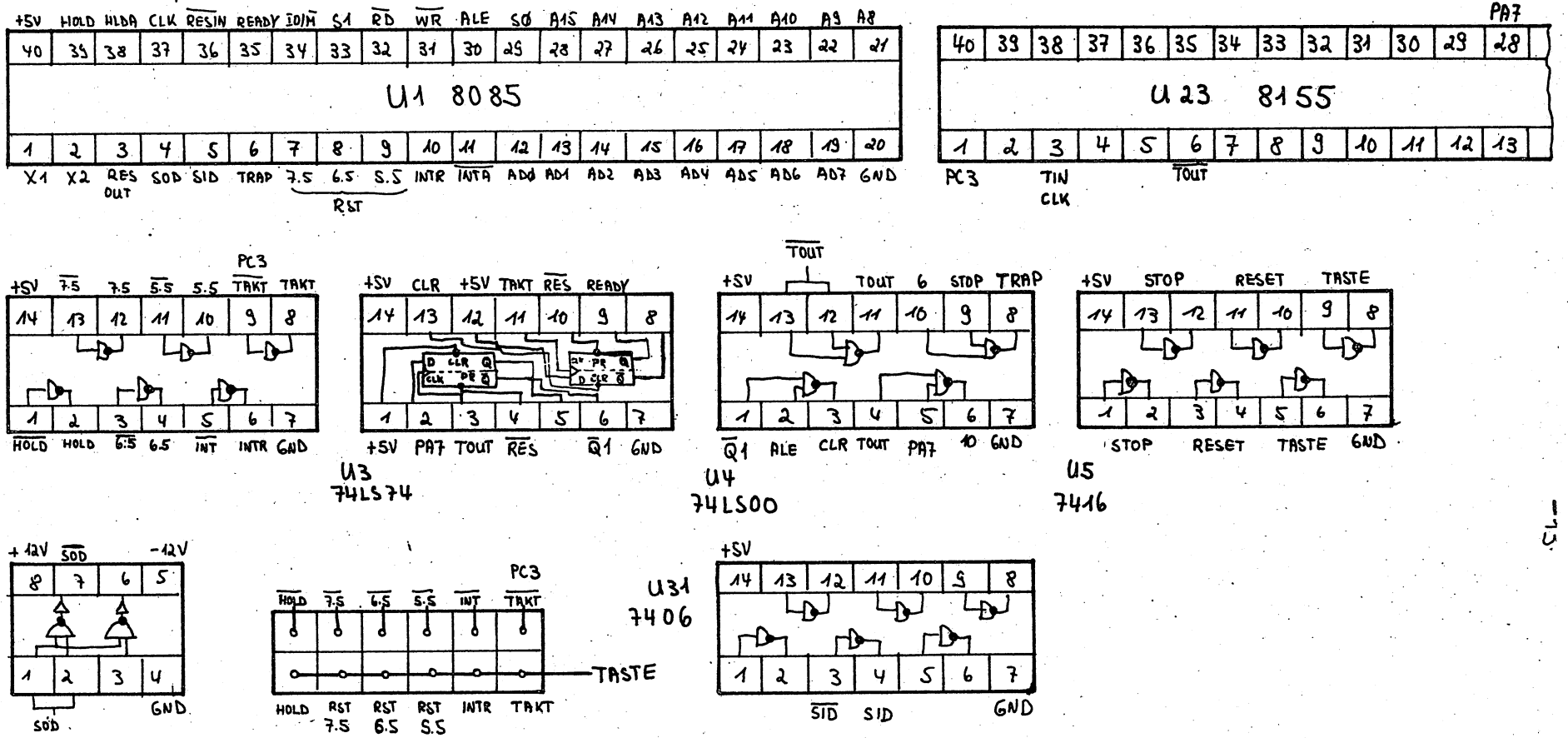


Bild 7: Signalplan Prozessorsteuerung

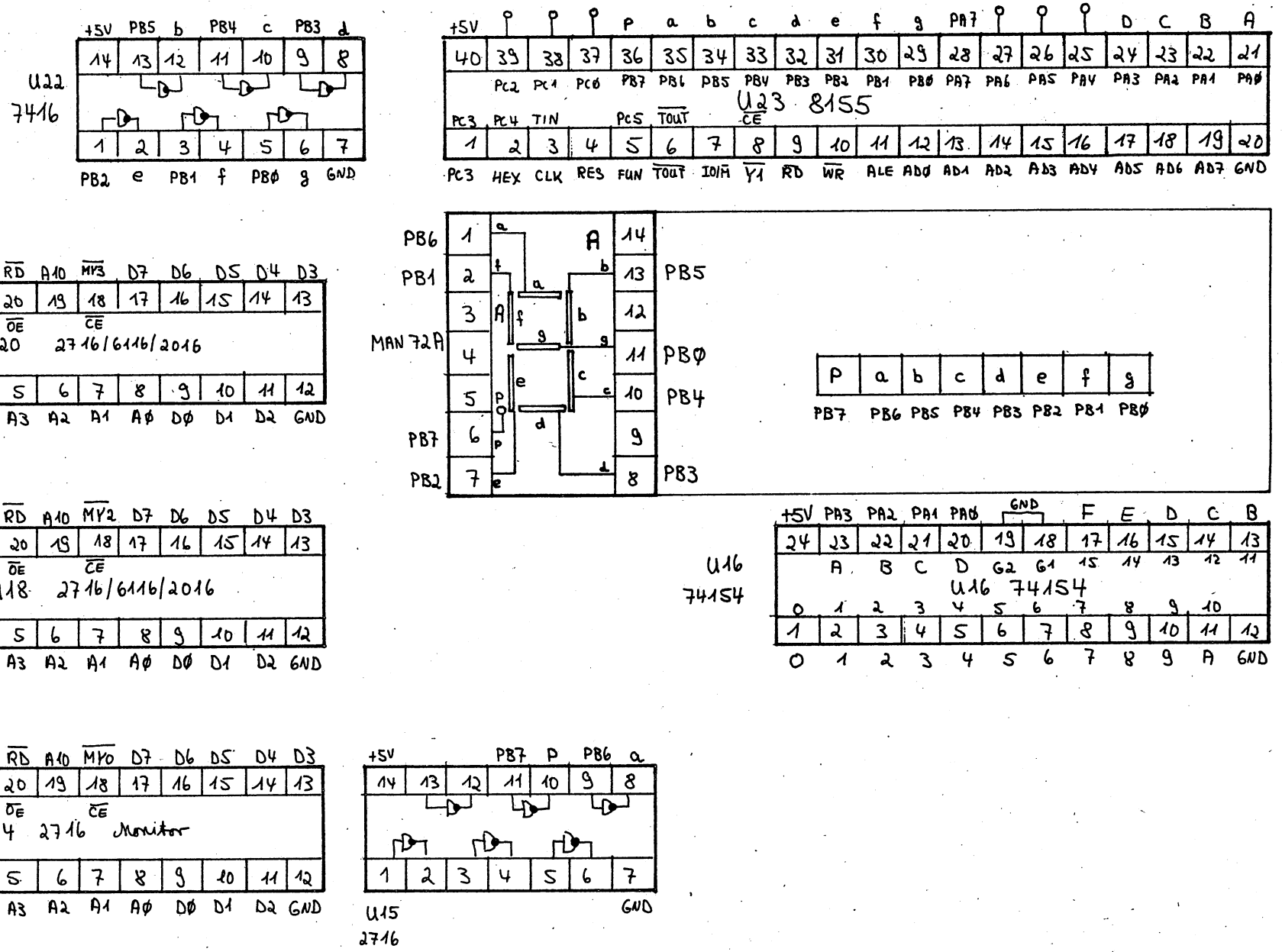


Bild 8: Signalplan Ein/Ausgabeinheit und Speicherbausteine

Bauteile MVUS 8085

1 Platine gebohrt und verzinnt, unbestückt
1 Netzteil nach besonderen Angaben
1 Verbindungskabel zum POWERACE-Steckpult nach bes. Angaben
1 Sockel 8 pol.
25 Sockel 14 pol.
8 Sockel 16 pol.
6 Sockel 20 pol.
4 Sockel 24 pol.
3 Sockel 40 pol.
1 Programmiersockel 24 pol.
1 Mikroprozessor 8085A
1 Baustein 8155 RAM/IO/TIMER
1 Parallelschnittstelle 8255
1 EPROM 2716 mind. 450 ns
2 RAM 6116 bzw. 2016 pinkompatibel mit EPROM 2716
1 74LS14 6 invertierende Schmitt-Trigger
1 74LS74 2 D-Flip-Flop
1 74LS00 4 NAND
3 7416 6 invertierende Treiber o.C.
1 7406 6 invertierende Treiber o.C. für 30 Volt
3 74LS245 8 Bustreiber
1 74LS241 8 Bustreiber
1 74LS240 8 invertierende Bustreiber
1 74LS373 8 D-Latch
1 74LS02 4 NOR
1 74LS32 4 OR
1 74LS08 4 AND
1 74LS155 2 1-aus-4-Decoder
1 74154 1-aus-16-Decoder
6 9370 Latch und HEX-Decoder (Fairchild)
11 BC327 pnp-Transistor (ersatzweise BC212)
16 rote Siebensegmentanzeige gem. Anode (ähnl. MAN 72A)
1 Schiebeschalter 8pol für 16pol. Sockel
27 Taster Schadow Umschalter 17 X 17 mm davon
16 weiß
8 blau
3 rot
1 Quarz 4 MHz Gehäuse 10X13 mm
4 LED 3 mm rot
2 LED 3 mm grün
2 LED 3 mm gelb
1 Steckverbinder 31 pol. Buchse
1 Z-Diode BZX 5.6
1 Z-Diode ZPD 4.7
1 Diode 1N4148

Alle Widerstände 1/4W 5%
8 56 Ohm
48 390 Ohm
11 1 KOhm
1 2.2 KOhm
1 3 KOhm
20 4.7 kOHM
3 10 kOhm
1 Widerstandsnetzwerk gereiht 8x4.7 KOhm

2 Elko 22uF/25V
6 Kondensator 0.1 uF
2 Kondensator 5.2 pF
1 Pfostenreihe einzeilig 20 Pfosten
1 Pfostenreihe doppelzeilig 6 Paare
5 Brücken dazu

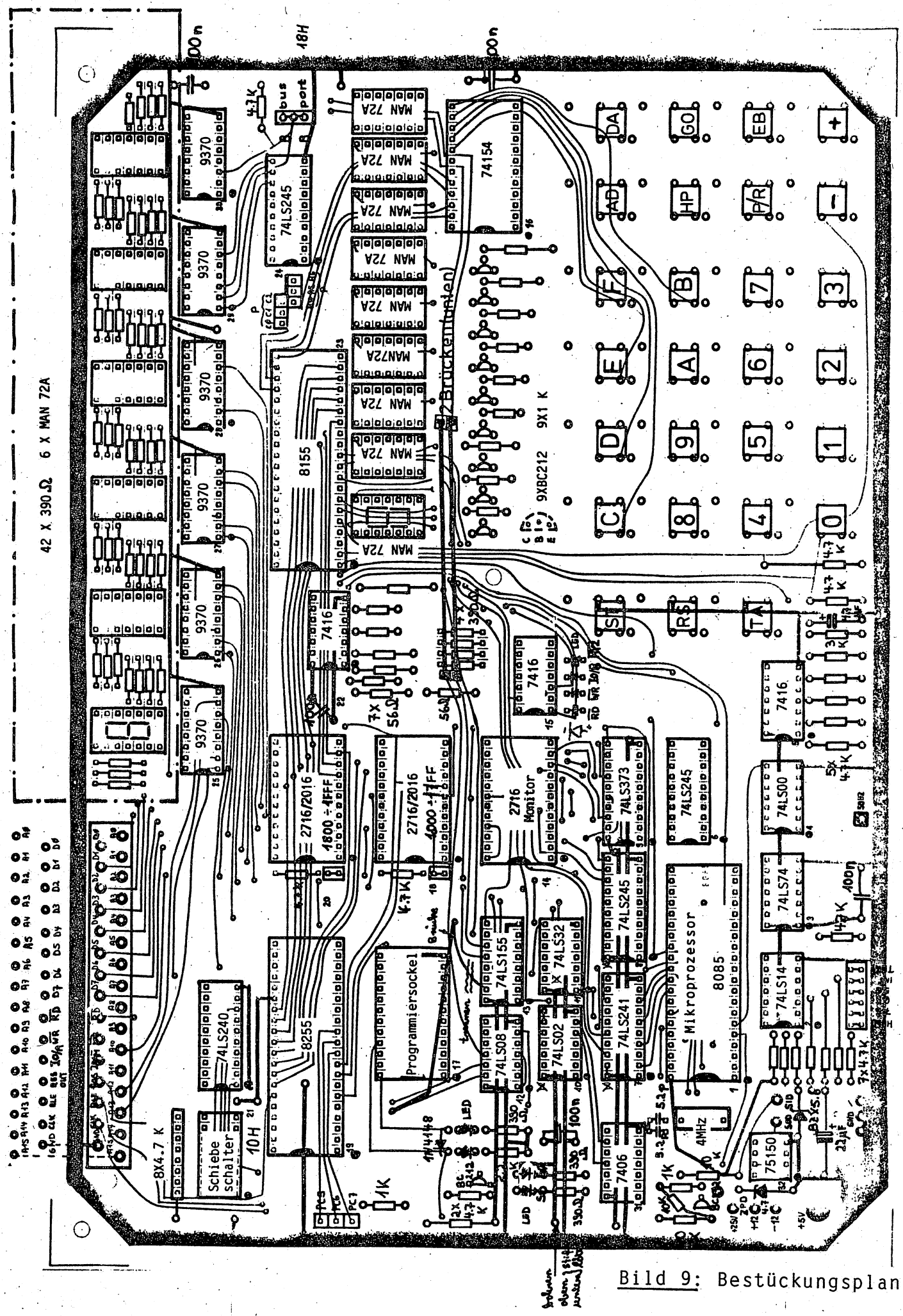


Bild 9: Bestückungsplan

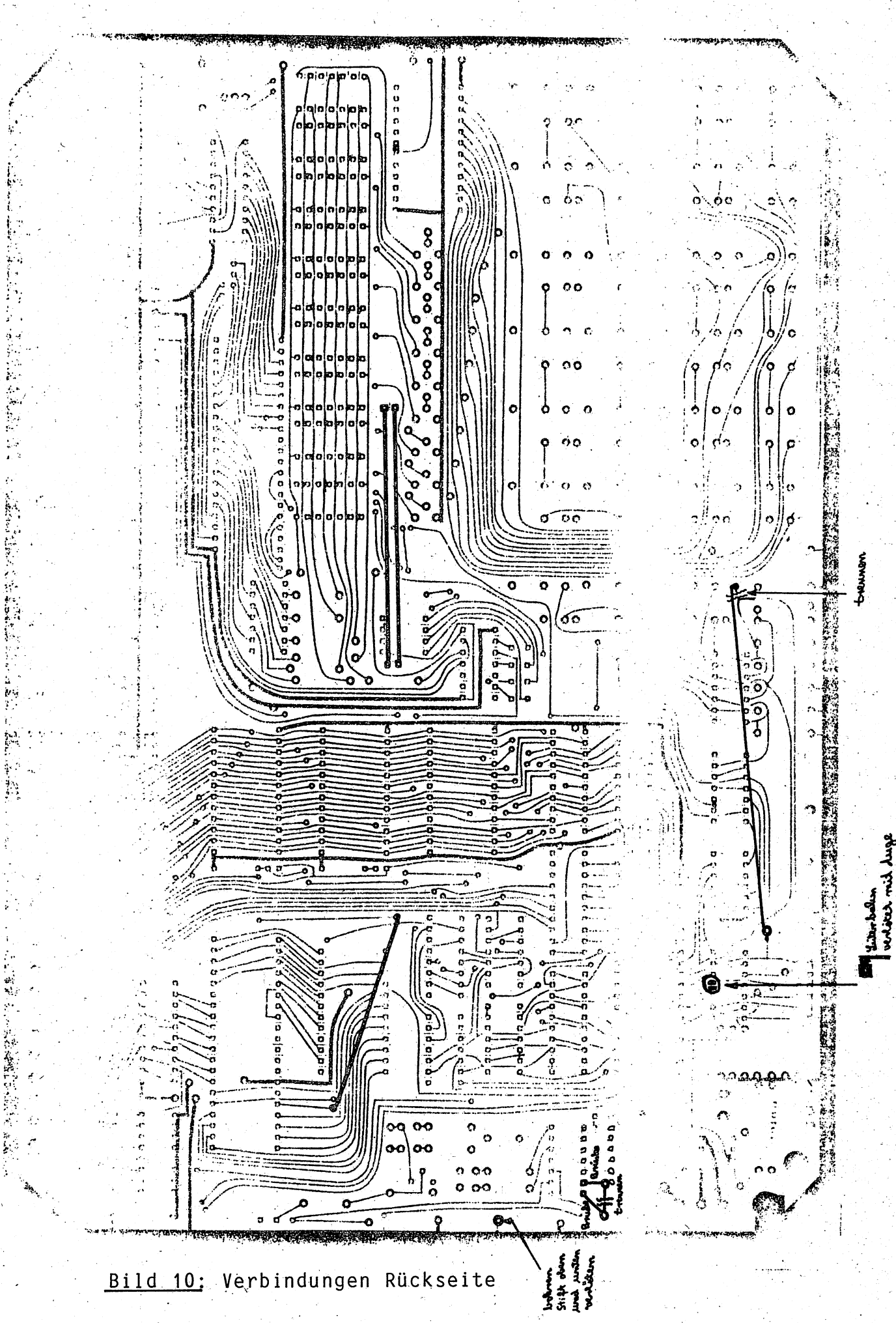


Bild 10: Verbindungen Rückseite

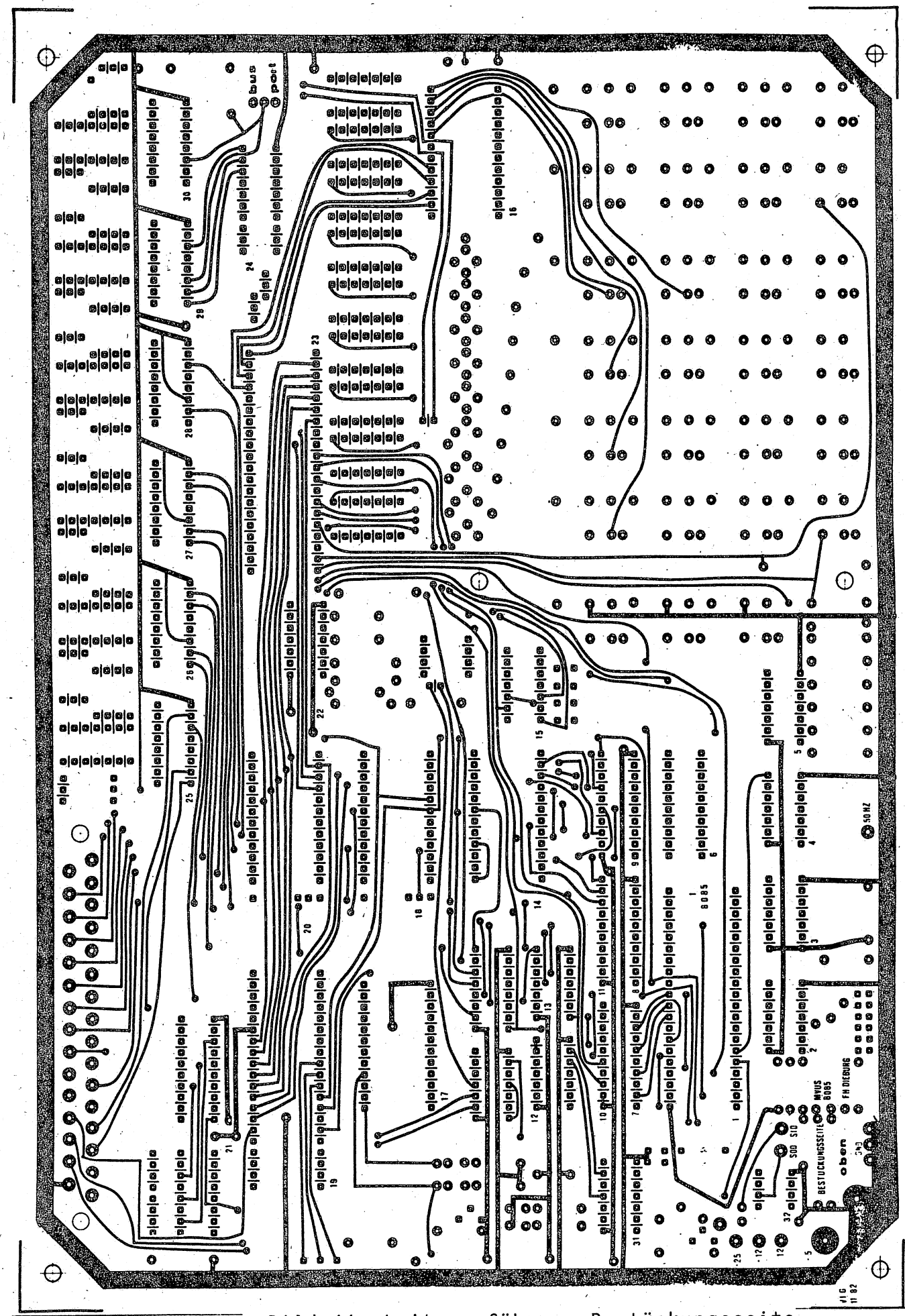


Bild 11: Leitungsführung Bestückungsseite

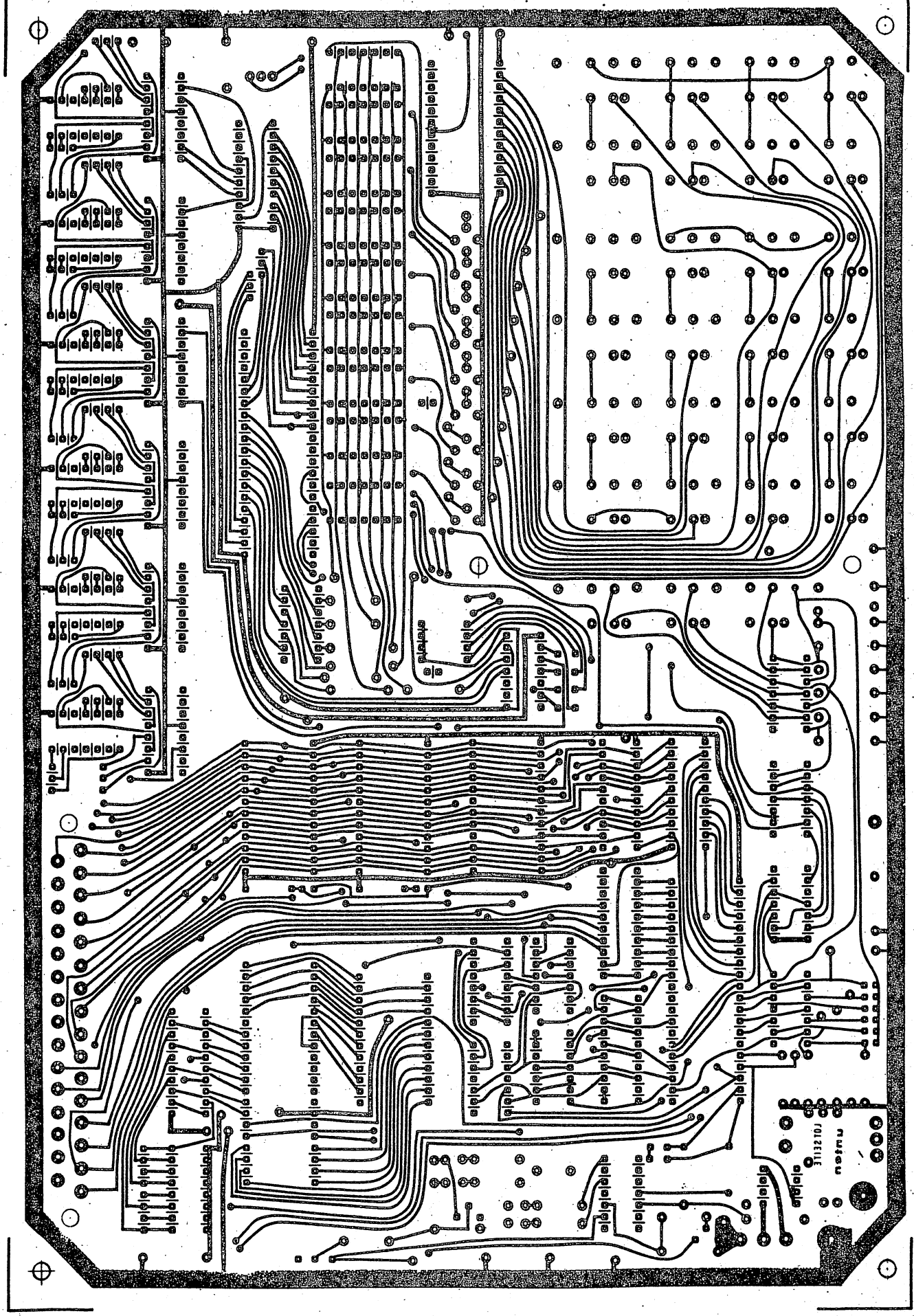


Bild 12: Leitungsführung Lötseite

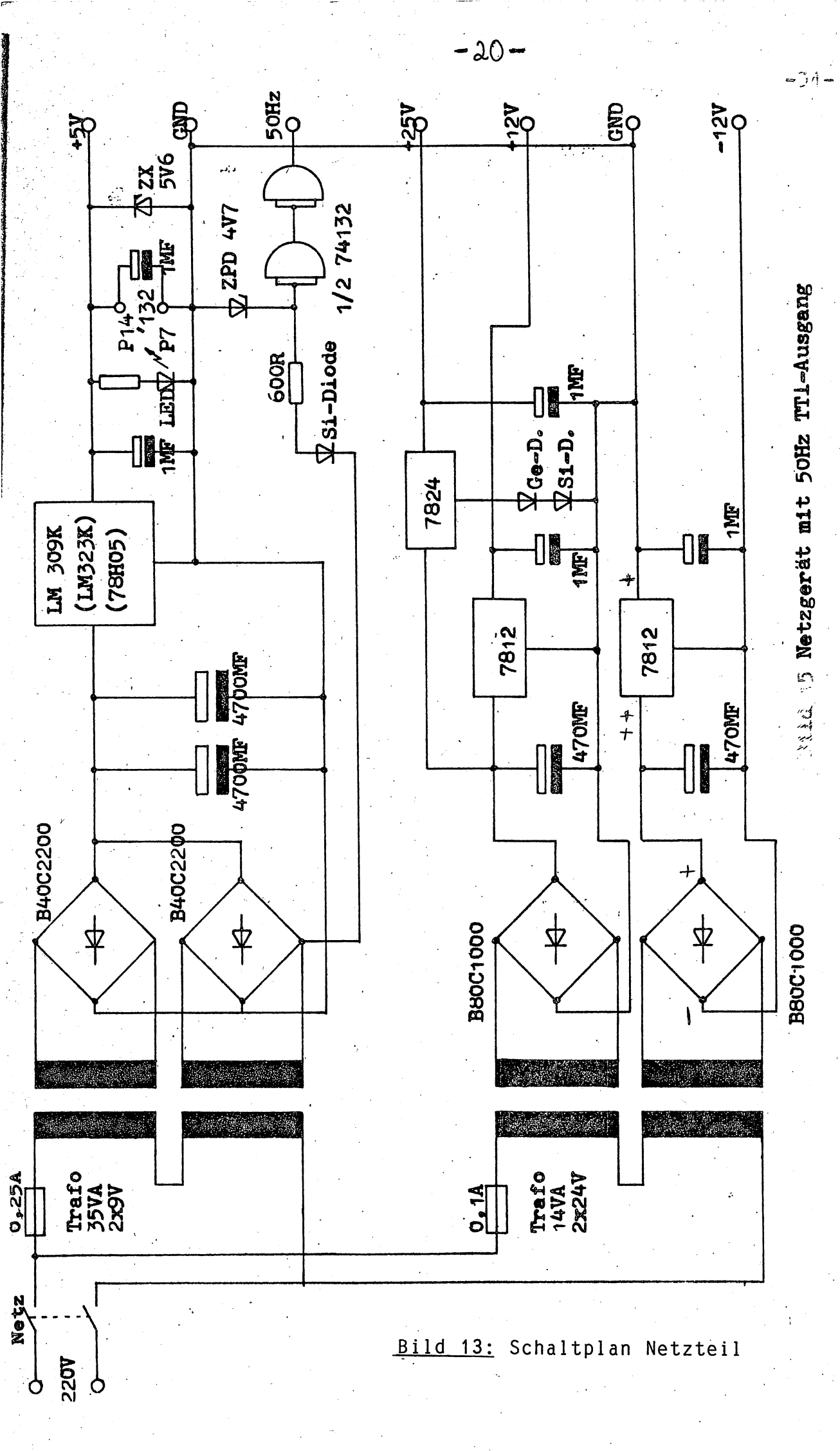
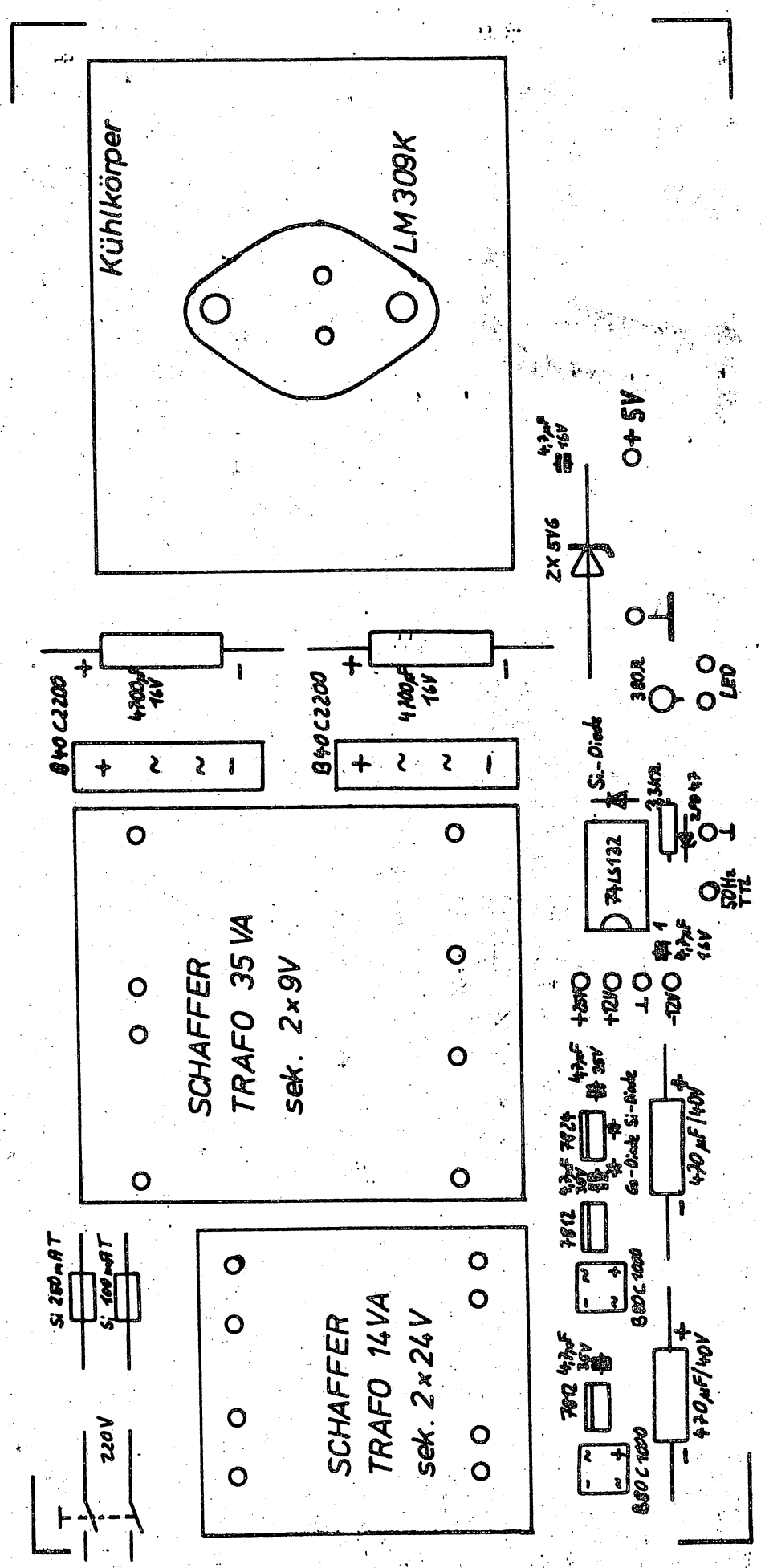


Bild 13: Schaltplan Netzteil

Bauteile Netzgerät MVUS 8085 5V/4A +25V,+12V,-12V/0.2A 50Hz TTL

1	Trafo 2X110/2X9V 35VA KLF 35 Schaffer
1	Trafo 2X110/2X24V 14VA KLF 14 Schaffer
2	Gleichrichter B40C3300
2	Gleichrichter B80C800
2	Diode 1N4148
1	Stabi 78H05
2	Stabi 7812 TO220
1	Stabi 7824 TO220
1	74LS132 Schmitt-Trigger
1	14pol Sockel
2	Elko 4700/16V
3	Elko 470/40V
2	Tantalelko 4,7/16V
3	Tantalelko 4,7/35V
1	Widerstand 3.3 KOhm
1	Z-Diode ZPD4.7
1	Kühlkörper SKOV 75/TO3/5A
1	Kaltgeräteeinheit
1	Kaltgerätestecker mit Schnur

BESTÜCKUNGSPLAN NETZTEIL



```
                        0001 >; 8085 - TASTEN - MONITOR
                        0002 >; VERSION 24.6.82  GEAENDERT 8.1.83  P/R -  E  VERT.
                        0003 >; 8155 - REGISTER
*8000                   0004 >SREG    EQU   08H       ; STATUS/STEUERREGISTER
*8000                   0005 >APORT   EQU   09H       ; A-REGISTER 1-AUS-16-DECODER
*8000                   0006 >BPORT   EQU   0AH       ; B-REGISTER 7-SEGMENT-CODE
*8000                   0007 >CPORT   EQU   0BH       ; C-REGISTER TASTENZEILEN
*8000                   0008 >TLOW    EQU   0CH       ; TIMER LOW
*8000                   0009 >THIGH   EQU   0DH       ; TIMER HIGH
                        000A >; RST- UND INTERRUPT-PROGRAMME SPRUENGE NACH MONITOR-RAM
#0000                   000B >        ORG   0000H     ; E P R O M
*0000 00                000C >RESET:  NOP             ;
*0001 F3                000D >        DI              ;
*0002 31 D0 08          000E >        LXI   SP,MSTA   ; MONITOR - STAPELZEIGER
*0005 C3 40 00          000F >        JMP   START     ; MONITOR - STARTADRESSE
*0008 F3                0010 >        DI              ; RST 1
*0009 C3 08 08          0011 >        JMP   RST1      ;
*000C FF FF             0012 >        DW    0FFFFH    ; RESERVE
*000E FF FF             0013 >        DW    0FFFFH    ; RESERVE
*0010 F3                0014 >        DI              ; RST 2
*0011 C3 10 08          0015 >        JMP   RST2      ;
*0014 FF FF             0016 >        DW    0FFFFH    ; RESERVE
*0016 FF FF             0017 >        DW    0FFFFH    ; RESERVE
*0018 F3                0018 >        DI              ; RST 3
*0019 C3 18 08          0019 >        JMP   RST3      ;
*001C FF FF             001A >        DW    0FFFFH    ; RESERVE
*001E FF FF             001B >        DW    0FFFFH    ; RESERVE
*0020 F3                001C >        DI              ; RST 4
*0021 C3 20 08          001D >        JMP   RST4      ;
*0024 F3                001E >        DI              ; TRAP - INTERRUPT
*0025 C3 24 08          001F >        JMP   TRAP      ;
*0028 F3                0020 >        DI              ; RST 5
*0029 C3 28 08          0021 >        JMP   RST5      ;
*002C F3                0022 >        DI              ; RST 5.5 INTERRUPT
*002D C3 2C 08          0023 >        JMP   RST55     ;
*0030 F3                0024 >        DI              ; RST 6
*0031 C3 30 08          0025 >        JMP   RST6      ;
*0034 F3                0026 >        DI              ; RST 6.5 INTERRUPT
*0035 C3 34 08          0027 >        JMP   RST65     ;
*0038 F3                0028 >        DI              ; RST 7 = HALTEPUNKT
*0039 C3 38 08          0029 >        JMP   RST7      ;
*003C F3                002A >        DI              ; RST 7.5 INTERRUPT
*003D C3 3C 08          002B >        JMP   RST75     ;
```

```
                        002C >; MONITOR - KALTSTART - ADRESSE
*0040 21 00 08  002D >START:  LXI   H,RSTO    ; SYSTEM - RAM - ANFANG
*0043 11 00 09  002E >        LXI   D,REND    ; SYSTEM - RAM - ENDE
*0046 36 00     002F >MON1:   MVI   M,00H     ; RAM LOESCHEN
*0048 23        0030 >        INX   H         ;
*0049 7C        0031 >        MOV   A,H       ; ENDE PRUEFEN
*004A BA        0032 >        CMP   D         ;
*004B C2 46 00  0033 >        JNZ   MON1      ; NEIN
*004E 7D        0034 >        MOV   A,L       ;
*004F BB        0035 >        CMP   E         ;
*0050 C2 46 00  0036 >        JNZ   MON1      ; NEIN
                0037 >; INTERRUPT - PROGRAMME - VORBESETZEN
*0053 06 10     0038 >        MVI   B,16      ; ZAEHLER
*0055 21 00 08  0039 >        LXI   H,RSTO    ; ANFANG DER INTERRUPTPROGRAMME
*0058 11 40 00  003A >        LXI   D,START   ; MONITOR-KATLSTART-ADRESSE
*005B 36 00     003B >MON2:   MVI   M,00H     ; NOP - BEFEHL
*005D 23        003C >        INX   H         ;
*005E 36 C3     003D >        MVI   M,0C3H    ; JMP - BEFEHL
*0060 23        003E >        INX   H         ;
*0061 73        003F >        MOV   M,E       ; SPRUNGADRESSE LOW
*0062 23        0040 >        INX   H         ;
*0063 72        0041 >        MOV   M,D       ; SPRUNGADRESSE HIGH
*0064 23        0042 >        INX   H         ;
*0065 05        0043 >        DCR   B         ; ZAEHLER - 1
*0066 C2 5B 00  0044 >        JNZ   MON2      ;


                0045 >; MONITOR - WARMSTART - ADRESSE
*0069 31 D0 08  0046 >MONW:   LXI   SP,MSTA   ; MONITOR - STAPELZEIEGR
                0047 >; BENUTZER - ANFANGSWERTE
*006C 21 00 10  0048 >        LXI   H,URAM    ; ANFANG BENUTZER-RAM-BEREICH
*006F 22 D0 08  0049 >        SHLD  LADD      ; LAUFENDE EINGABEADRESSE
*0072 22 D4 08  004A >        SHLD  PCR       ; BENUTZER-BEFEHLSZAEHLER
*0075 21 A0 08  004B >        LXI   H,BSTA    ; BENUTZER-STAPEL-ADRESSE
*0078 22 D2 08  004C >        SHLD  BSP       ; BENUTZER-STAPELZEIGER
*007B 3E 1F     004D >        MVI   A,1FH     ; ALLE INTERRUPTS SPERREN
*007D 30        004E >        SIM             ; NACH INTERRUPTREGISTER
*007E 32 DE 08  004F >        STA   IREG      ; BENUTZER-INTERRUPT-REGISTER
                0050 >; SCHNITTSTELLE PROGRAMMIEREN UND BEGRUESSUNG
*0081 3E 03     0051 >        MVI   A,03H     ; STEUERBYTE A=AUS B=AUS C=EIN
*0083 D3 08     0052 >        OUT   SREG      ; STATUS/STEUERREGISTER
*0085 21 A0 07  0053 >        LXI   H,TEXT    ;  --8085--
*0088 11 EB 08  0054 >        LXI   D,AUS     ; AUSGABEPUFFER
*008B 06 09     0055 >        MVI   B,9       ; ZAEHLER
*008D 7E        0056 >MON3:   MOV   A,M       ; ANZEIGE UEBERTRAGEN
*008E 23        0057 >        INX   H         ;
*008F 12        0058 >        STAX  D         ;
*0090 13        0059 >        INX   D         ;
*0091 05        005A >        DCR   B         ; ZAEHLER - 1
*0092 C2 8D 00  005B >        JNZ   MON3      ;
                005C >;
```

```
                005D >; VERARBEITUNGSSCHLEIFE
                005E >;
*0095 CD BA 00  005F >MON5:   CALL  ANZ       ; ANZEIGE - UNTERPROGRAMM
*0098 DB 0B     0060 >        IN    CPORT     ; EINZEL-TAKT-TASTE ?
*009A E6 08     0061 >        ANI   08H       ; 00001000  C3 = 0 ?
*009C CA 20 03  0062 >        JZ    FUNET     : JA: EINZEL-TAKT MIT READY
*009F CD DB 00  0063 >        CALL  TEST      ; TASTATUR - ABFRAGE - UNTERPROGR
*00A2 D2 95 00  0064 >        JNC   MON5      ; KEINE TASTE BETAETIGT
*00A5 CD 81 01  0065 >        CALL  TASTE     ; TASTE AUSWERTEN
*00A8 C3 95 00  0066 >        JMP   MON5      ;
                0067 >;


                0068 >; MS1 = N * 1 MS WARTEN ; N IN AKKU UEBERGEBEN
*00AB B7        0069 >MS1:    ORA   A         ;    4 TAKTE : AKKU TESTEN
*00AC C8        006A >        RZ              ;    6 TAKTE   BEI RETURN 12
*00AD F5        006B >        PUSH  PSW       ;   12 TAKTE : AKKU RETTEN
*00AE 3E 8B     006C >        MVI   A,139     ;    7 TAKTE : ZAEHLER LADEN
*00B0 3D        006D >MS2:    DCR   A         ;              4 TAKTE
*00B1 C2 B0 00  006E >        JNZ   MS2       ; 1943 TAKTE = 14*139 - 3
*00B4 F1        006F >        POP   PSW       ;   10 TAKTE : AKKU ZURUECK
*00B5 00        0070 >        NOP             ;    4 TAKTE : AUSGLEICHEN
*00B6 3D        0071 >        DCR   A         ; N = N - 1
*00B7 C3 AB 00  0072 >        JMP   MS1       ;   10 TAKTE : SCHLEIFE
                0073 >; FUER EINEN DURCHLAUF  : 2000 TAKTE = 1000 US
                0074 >; ANZ = AUSGABEPUFFER  AUS  ANZEIGEN
*00BA F5        0075 >ANZ:    PUSH  PSW       ; AKKU UND PSW RETTEN
*00BB C5        0076 >        PUSH  B         ; BC RETTEN
*00BC E5        0077 >        PUSH  H         ; HL RETTEN
*00BD 21 EB 08  0078 >        LXI   H,AUS     ; ADRESSE AUSGABEPUFFER
*00C0 01 00 09  0079 >        LXI   B,0900H   ; ZAEHLER UND DIGITADRESSE
*00C3 79        007A >ANZ1:   MOV   A,C       ; DIGITADRESSE
*00C4 D3 09     007B >        OUT   APORT     ; AUSGEBEN
*00C6 7E        007C >        MOV   A,M       ; 7-SEGMENT-CODE
*00C7 D3 0A     007D >        OUT   BPORT     ; AUSGEBEN
*00C9 23        007E >        INX   H         ; NEUE STELLE
*00CA 0C        007F >        INR   C         ; NEUE STELLE
*00CB 3E 02     0080 >        MVI   A,2H      ; 2 MS WARTEN
*00CD CD AB 00  0081 >        CALL  MS1       ; UNTERPROGRAMM
*00D0 AF        0082 >        XRA   A         ; WAEHREND DES UMSCHALTENS
*00D1 D3 0A     0083 >        OUT   BPORT     ; DUNKEL TASTEN
*00D3 05        0084 >        DCR   B         ; DIGITZAEHLER - 1
*00D4 C2 C3 00  0085 >        JNZ   ANZ1      ;
*00D7 E1        0086 >        POP   H         ; HL ZURUECK
*00D8 C1        0087 >        POP   B         ; BC ZURUECK
*00D9 F1        0088 >        POP   PSW       ; AKKU UND PSW ZURUECK
*00DA C9        0089 >        RET             : RUECKSPRUNG
```

```
                    008A >; TEST = TASTEN PRUEFEN
                    008B >; JA:  CARRY = 1  A=ZEILE  B=SPALTE
*00DB 06 0F         008C >TEST:   MVI   B,0FH      ; SPALTENZAEHLER
*00DD 78            008D >TEST1:  MOV   A,B        ; NACH AKKU
*00DE D3 09         008E >        OUT   APORT      ; AUSGEBEN
*00E0 DB 0B         008F >        IN    CPORT      ; ZEILE LESEN
*00E2 E6 30         0090 >        ANI   30H        ; MASKE XX??XXXX = 00110000
*00E4 FE 30         0091 >        CPI   30H        ; BEIDE ZEILEN HIGH ?
*00E6 C2 EF 00      0092 >        JNZ   TEST2      ; TASTE GEDRUECKT
*00E9 05            0093 >        DCR   B          ; NEUE SPALTE
*00EA F2 DD 00      0094 >        JP    TEST1      ; >= NULL
*00ED B7            0095 >        ORA   A          ; CARRY = 0: KEINE TASTE
*00EE C9            0096 >        RET              ; RUECKSPRUNG
*00EF 37            0097 >TEST2:  STC              ; CARRY = 1: TASTE GEDRUECKT
*00F0 C9            0098 >        RET              ; RUECKSPRUNG
                    0099 >; CODL = LINKES HALBBYTE UMCODIEREN
*00F1 0F            009A >CODL:   RRC              ; 1 BIT RECHTS
*00F2 0F            009B >        RRC              ; 1 BIT RECHTS
*00F3 0F            009C >        RRC              ; 1 BIT RECHTS
*00F4 0F            009D >        RRC              ; 1 BIT RECHTS
                    009E >; CODR = RECHTS HALBBYTE UMCODIEREN
*00F5 C5            009F >CODR:   PUSH  B          ; BC RETTEN
*00F6 E5            00A0 >        PUSH  H          ; HL RETTEN
*00F7 E6 0F         00A1 >        ANI   0FH        ; LINKES HALBBYTE MASKIEREN
*00F9 06 00         00A2 >        MVI   B,0        ; B LOESCHEN
*00FB 4F            00A3 >        MOV   C,A        ; BITS NACH C
*00FC 21 A9 07      00A4 >        LXI   H,COTAB    ; ANFANGSADRESSE CODETABELLE
*00FF 09            00A5 >        DAD   B          ; BITMUSTER DAZU
*0100 7E            00A6 >        MOV   A,M        ; 7-SEGMENTCODE LADEN
*0101 E1            00A7 >        POP   H          ; HL ZURUECK
*0102 C1            00A8 >        POP   B          ; BC ZURUECK
*0103 C9            00A9 >        RET              ; RUECKSPRUNG
                    00AA >; AUSHL = INHALT VON HL UND ADRESSIERTES BYTE ANZEIGEN
*0104 3E 01         00AB >AUSHL:  MVI   A,01H      ; -
*0106 32 F0 08      00AC >        STA   AUS+5      ;
*0109 32 F3 08      00AD >        STA   AUS+8      ;
*010C 7C            00AE >        MOV   A,H        ; ADRESSE HIGH
*010D F5            00AF >        PUSH  PSW        ;
*010E CD F1 00      00B0 >        CALL  CODL       ; UMWANDELN
*0111 32 EC 08      00B1 >        STA   AUS+1      ;
*0114 F1            00B2 >        POP   PSW        ;
*0115 CD F5 00      00B3 >        CALL  CODR       ; UMWANDELN
*0118 32 ED 08      00B4 >        STA   AUS+2      ;
*011B 7D            00B5 >        MOV   A,L        ; ADRESSE LOW
*011C F5            00B6 >        PUSH  PSW        ;
*011D CD F1 00      00B7 >        CALL  CODL       ; UMWANDELN
*0120 32 EE 08      00B8 >        STA   AUS+3      ;
*0123 F1            00B9 >        POP   PSW        ;
*0124 CD F5 00      00BA >        CALL  CODR       ; UMWANDELN
*0127 32 EF 08      00BB >        STA   AUS+4      ;
                    00BC >; EINSPRUNGPUNKT AKKUMULATORANZEIGE
*012A 7E            00BD >AUSHL1: MOV   A,M        ; INHALT
*012B F5            00BE >        PUSH  PSW        ;
*012C CD F1 00      00BF >        CALL  CODL       ; UMWANDELN
*012F 32 F1 08      00C0 >        STA   AUS+6      ;
*0132 F1            00C1 >        POP   PSW        ;
*0133 CD F5 00      00C2 >        CALL  CODR       ; UMWANDELN
*0136 32 F2 08      00C3 >        STA   AUS+7      ;
                    00C4 >; WARTE = ANZEIGEN UND AUF TASTENFREIGABE WARTEN
*0139 CD BA 00      00C5 >WARTE:  CALL  ANZ        ; ANZEIGEZYKLUS
*013C CD DB 00      00C6 >        CALL  TEST       ; TASTATURABFRAGE
*013F DA 39 01      00C7 >        JC    WARTE      ; TASTE NOCH GEDRUECKT
*0142 C9            00C8 >        RET              ; TASTE FREIGEGEBEN
```

```
                        00C9 >; INBY = B-REGISTER IN DAS (HL)-BYTE EINFUEGEN
*0143 7E                00CA >INBY:   MOV   A,M       ; ALTES BYTE HOLEN
*0144 07                00CB >        RLC             ; 1 BIT LINKS
*0145 07                00CC >        RLC             ; 1 BIT LINKS
*0146 07                00CD >        RLC             ; 1 BIT LINKS
*0147 07                00CE >        RLC             ; 1 BIT LINKS
*0148 E6 F0             00CF >        ANI   0F0H      ; STELLE MASKIEREN
*014A B0                00D0 >        ORA   B         ; NEUE STELLE DAZU
*014B 77                00D1 >        MOV   M,A       ; ABSPEICHERN
*014C C9                00D2 >        RET             ;
                        00D3 >; INWD = B-REGISTER IN DAS (DE)-WORT EINFUEGEN
                        00D4 >; HL ENTHAELT NEUEN INHALT FUER AUSHL  DE BLEIBT
*014D 1A                00D5 >INWD:   LDAX  D         ; WORT LOW HOLEN
*014E 13                00D6 >        INX   D         ;
*014F 6F                00D7 >        MOV   L,A       ; NACH L
*0150 1A                00D8 >        LDAX  D         ; WORT HIGH HOLEN
*0151 67                00D9 >        MOV   H,A       ; NACH H
*0152 29                00DA >        DAD   H         ; 1 BIT LINKS
*0153 29                00DB >        DAD   H         ; 1 BIT LINKS
*0154 29                00DC >        DAD   H         ; 1 BIT LINKS
*0155 29                00DD >        DAD   H         ; 1 BIT LINKS
*0156 48                00DE >        MOV   C,B       ; NEUE STELLE
*0157 06 00             00DF >        MVI   B,00H     ;
*0159 09                00E0 >        DAD   B         ; RECHTS DAZU
*015A 7C                00E1 >        MOV   A,H       ; NEUES WORT HIGH
*015B 12                00E2 >        STAX  D         ; SPEICHERN
*015C 1B                00E3 >        DCX   D         ;
*015D 7D                00E4 >        MOV   A,L       ; NEUES WORT LOW
*015E 12                00E5 >        STAX  D         ;
*015F C9                00E6 >        RET             ;
                        00E7 >; TABELLEN DURCHSUCHEN
                        00E8 >; HPSU = HALTEPUNKT IN  SYHP  SUCHEN  ADRESSE IN HL
*0160 21 F0 07          00E9 >HPSU:   LXI   H,SYHP+3  ; TABELLENANFANG
*0163 16 04             00EA >        MVI   D,4       ; 4 HALTEPUNKTE
*0165 BE                00EB >HPSU1:  CMP   M         ; MIT A VERGLEICHEN
*0166 C8                00EC >        RZ              ; GEFUNDEN
*0167 23                00ED >        INX   H         ;
*0168 23                00EE >        INX   H         ;
*0169 23                00EF >        INX   H         ;
*016A 15                00F0 >        DCR   D         ; ZAEHLER - 1
*016B C2 65 01          00F1 >        JNZ   HPSU1     ;
*016E 37                00F2 >        STC             ; NICHT GEFUNDEN
*016F C9                00F3 >        RET             ;
                        00F4 >; RESU = REGISTER IN  SYRE  SUCHEN  ADRESSE IN HL
*0170 21 CD 07          00F5 >RESU:   LXI   H,SYRE+4  ; TABELLENANFANG
*0173 16 07             00F6 >        MVI   D,7       ; 7 REGISTER
*0175 BE                00F7 >RESU1:  CMP   M         ; MIT A VERGLEICHEN
*0176 C8                00F8 >        RZ              ; GEFUNDEN
*0177 37                00F9 >        STC             ; C = 1: NICHT GEFUNDEN
*0178 15                00FA >        DCR   D         ; ZAEHLER - 1
*0179 C8                00FB >        RZ              ; ENDE DER LISTE
*017A 23                00FC >        INX   H         ;
*017B 23                00FD >        INX   H         ;
*017C 23                00FE >        INX   H         ;
*017D 23                00FF >        INX   H         ;
*017E C3 75 01          0100 >        JMP   RESU1     ; WEITER SUCHEN
```

```
                0101 >; TASTE = TASTE AUSWERTEN
*0181 FE 20     0102 >TASTE:  CPI   20H       ; HEXADEZIMALTASTE ?
*0183 C2 2F 02  0103 >        JNZ   TAST20    ; NEIN
                0104 >; H E X A D E Z I M A L T A S T E   B=TASTENWERT
*0186 3A EB 08  0105 >        LDA   AUS       ; STATUS ?
*0189 FE 3D     0106 >        CPI   3DH       ; D-STATUS ?
*018B C2 97 01  0107 >        JNZ   TAST1     ; NEIN
                0108 >; DATENEINGABE
*018E 2A D0 08  0109 >        LHLD  LADD      ; LAUFENDE EINGABEADRESSE
*0191 CD 43 01  010A >        CALL  INBY      ; NEUEN WERT EINBAUEN
*0194 C3 04 01  010B >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
*0197 FE 77     010C >TAST1:  CPI   77H       ; A-STATUS ?
*0199 C2 A5 01  010D >        JNZ   TAST2     ; NEIN
                010E >; ADRESSEINGABE
*019C 11 D0 08  010F >        LXI   D,LADD    ; ALTE EINGABEADRESSE
*019F CD 4D 01  0110 >        CALL  INWD      ; NEUEN WERT EINBAUEN
*01A2 C3 04 01  0111 >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
*01A5 FE 67     0112 >TAST2:  CPI   67H       ; P-STATUS ?
*01A7 C2 B3 01  0113 >        JNZ   TAST3     ; NEIN
                0114 >; BEFEHLSZAEHLEREINGABE
*01AA 11 D4 08  0115 >        LXI   D,PCR     ; ALTEN BEFEHLSZAEHLER
*01AD CD 4D 01  0116 >        CALL  INWD      ; NEUEN WERT EINBAUEN
*01B0 C3 04 01  0117 >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
*01B3 FE 05     0118 >TAST3:  CPI   05H       ; R-STATUS ?
*01B5 C2 1E 02  0119 >        JNZ   TAST12    ; NEIN
*01B8 3A EC 08  011A >        LDA   AUS+1     ; REGISTERART ?
*01BB FE 86     011B >        CPI   086H      ; .I INTERRUPT-REGISTER ?
*01BD CA 39 01  011C >        JZ    WARTE     ; KEINE EINGABE
*01C0 FE 7E     011D >        CPI   7EH       ; 0 BEI BEDINGUNSREGISTER
*01C2 CA 39 01  011E >        JZ    WARTE     ; KEINE EINGABE
*01C5 FE 30     011F >        CPI   30H       ; 1 BEI BEDINGUNGSREGISTER
*01C7 CA 39 01  0120 >        JZ    WARTE     ; KEINE EINGABE
*01CA FE 77     0121 >        CPI   77H       ; A-REGISTER ?
*01CC C2 E7 01  0122 >        JNZ   TAST5     ; NEIN
                0123 >; AKKUMULATOREINGABE
*01CF 21 D7 08  0124 >        LXI   H,AKKU    ; ALTEN INHALT
*01D2 CD 43 01  0125 >        CALL  INBY      ; NEUEN WERT EINBAUEN
                0126 >; EINSPRUNGPUNKT AKKUMULATORSTATUS
*01D5 21 01 01  0127 >TAST4:  LXI   H,0101H   ;
*01D8 22 EE 08  0128 >        SHLD  AUS+3     ; - -
*01DB 22 F0 08  0129 >        SHLD  AUS+5     ; - -
*01DE 22 F2 08  012A >        SHLD  AUS+7     ; - -
*01E1 21 D7 08  012B >        LXI   H,AKKU    ; ADRESSE AKKUMULATOR
*01E4 C3 2A 01  012C >        JMP   AUSHL1    ; ANZEIGEN UND WARTEN
*01E7 CD 70 01  012D >TAST5:  CALL  RESU      ; REGISTER SUCHEN
*01EA DA 39 01  012E >        JC    WARTE     ; NICHT GEFUNDEN
*01ED 23        012F >        INX   H         ;
*01EE 23        0130 >        INX   H         ;
*01EF 5E        0131 >        MOV   E,M       ; ADRESSE LOW
*01F0 23        0132 >        INX   H         ;
*01F1 56        0133 >        MOV   D,M       ; ADRESSE HIGH
*01F2 CD 4D 01  0134 >        CALL  INWD      ; NEUE STELLE EINFUEGEN
                0135 >; EINSPRUNGPUNKT DC DE HL SP STATUS INHALT IN HL
*01F5 3E 01     0136 >TAST7:  MVI   A,01H     ;
*01F7 32 EE 08  0137 >        STA   AUS+3     ;  -
*01FA 32 F3 08  0138 >        STA   AUS+8     ;  -
*01FD 7C        0139 >        MOV   A,H       ; H-ADRESSE
*01FE F5        013A >        PUSH  PSW       ;
*01FF CD F1 00  013B >        CALL  CODL      ; UMWANDELN
```

```
*0202 32 EF 08  013C >          STA   AUS+4       ;
*0205 F1        013D >          POP   PSW         ;
*0206 CD F5 00  013E >          CALL  CODR        ; UMWANDELN
*0209 32 F0 08  013F >          STA   AUS+5       ;
*020C 7D        0140 >          MOV   A,L         ; L-ADRESSE
*020D F5        0141 >          PUSH  PSW         ;
*020E CD F1 00  0142 >          CALL  CODL        ; UMWANDELN
*0211 32 F1 08  0143 >          STA   AUS+6       ;
*0214 F1        0144 >          POP   PSW         ;
*0215 CD F5 00  0145 >          CALL  CODR        ; UMWANDELN
*0218 32 F2 08  0146 >          STA   AUS+7       ;
*021B C3 39 01  0147 >          JMP   WARTE       ; ANZEIGEN UND WARTEN
                0148 >; HALTEPUNKTEINGABE PRUEFEN
*021E B7        0149 >TAST12:   ORA   A           ; HALTEPUNKTEINGABE ?
*021F F2 39 01  014A >          JP    WARTE       ; NEIN
*0222 CD 60 01  014B >          CALL  HPSU        ; HALTEPUNKT SUCHEN
*0225 23        014C >          INX   H           ;
*0226 5E        014D >          MOV   E,M         ; ADRESSE LOW
*0227 23        014E >          INX   H           ;
*0228 56        014F >          MOV   D,M         ; ADRESSE HIGH
*0229 CD 4D 01  0150 >          CALL  INWD        ; NEUEN WERT EINBAUEN
*022C C3 04 01  0151 >          JMP   AUSHL       ; ANZEIGEN UND WARTEN
                0152 >;

                0153 >; F U N K T I O N S T A S T E N - SPRUNGVERTEILER
*022F 68        0154 >TAST20:   MOV   L,B         ; TASTENWERT
*0230 26 00     0155 >          MVI   H,0         ; NACH HL
*0232 29        0156 >          DAD   H           ; MAL 2
*0233 01 B9 07  0157 >          LXI   B,FUTAB     ; ADRESSE FUNKTIONSTABELLE
*0236 09        0158 >          DAD   B           ; ADDIEREN
*0237 5E        0159 >          MOV   E,M         ; ADRESSE LOW
*0238 23        015A >          INX   H           ;
*0239 56        015B >          MOV   D,M         ; ADRESSE HIGH
*023A EB        015C >          XCHG              ; VERTAUSCHEN
*023B E9        015D ->         PCHL              ; INDIZIERTER SPRUNG


                015E >; A-TASTE  ADRESSEINGABEBETRIEB
*023C 3E 77     015F >FUNAD:    MVI   A,77H       ; A
*023E 32 EB 08  0160 >FUNAD1:   STA   AUS         ; STATUS
*0241 2A D0 08  0161 >          LHLD  LADD        ; LAUFENDE EINGABEADRESSE
*0244 C3 04 01  0162 >          JMP   AUSHL       ; ANZEIGEN UND WARTEN
                0163 >; D-TASTE  DATENEINGABEBETRIEB
*0247 3E 3D     0164 >FUNDA:    MVI   A,3DH       ; D
*0249 C3 3E 02  0165 >          JMP   FUNAD1      ;
                0166 >; P/R-TASTE  REGISTEREINGABEBETRIEB
*024C 3A EB 08  0167 >FUNPC:    LDA   AUS         ; ALTEN STATUS PRUEFEN
*024F FE 67     0168 >          CPI   67H         ; P ?
*0251 CA 5F 02  0169 >          JZ    FUNPC1      ; JA
*0254 3E 67     016A >          MVI   A,67H       ; AUS R WIRD P
*0256 32 EB 08  016B >          STA   AUS         ; NEUER STATUS
*0259 2A D4 08  016C >          LHLD  PCR         ; BEFEHLSZAEHLER
*025C C3 04 01  016D >          JMP   AUSHL       ; ANZEIGEN UND WARTEN
*025F 3E 05     016E >FUNPC1:   MVI   A,05H       ; AUS P WIRD R
*0261 32 EB 08  016F >          STA   AUS         ; NEUER STATUS
*0264 21 77 05  0170 >          LXI   H,0577H     ; rA
*0267 22 EC 08  0171 >          SHLD  AUS+1       ; REGISTERART
*026A C3 D5 01  0172 >          JMP   TAST4       ; ANZEIGEN UND WARTEN
                0173 >; H-TASTE  HALTEPUNKTEINGABE
*026D 3E FE     0174 >FUNHP:    MVI   A,0FEH      ; .0
*026F 32 EB 08  0175 >          STA   AUS         ; STATUS
*0272 2A DF 08  0176 >          LHLD  HPUA        ; HALTEPUNKTDARESSE
*0275 C3 04 01  0177 >          JMP   AUSHL       ; ANZEIGEN UND WARTEN
```

```
                    0178 >; - TASTE
*0278 01 FF FF  0179 >FUNMI:  LXI   B,0FFFFH  ; - 1
*027B C3 81 02  017A >        JMP   FUNP1     ;
                    017B >; + TASTE
*027E 01 01 00  017C >FUNPL:  LXI   B,0001    ; + 1
*0281 3A EB 08  017D >FUNP1:  LDA   AUS       ; STATUS PRUEFEN
*0284 FE 00     017E >        CPI   00H       ; LEER ?
*0286 CA 39 01  017F >        JZ    WARTE     ; KEINE WIRKUNG
*0289 FE 77     0180 >        CPI   77H       ; A-STATUS ?
*028B CA 93 02  0181 >        JZ    FUNP2     ; JA
*028E FE 3D     0182 >        CPI   3DH       ; D-STATUS ?
*0290 C2 9D 02  0183 >        JNZ   FUNP3     ; NEIN
                    0184 >; LAUFENDE EINGABEADRESSE LADD AENDERN
*0293 2A D0 08  0185 >FUNP2:  LHLD  LADD      ; LAUFENDE EINGABEADRESSE
*0296 09        0186 >        DAD   B         ;
*0297 22 D0 08  0187 >        SHLD  LADD      ; NEUER WERT
*029A C3 04 01  0188 >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
*029D FE 05     0189 >FUNP3:  CPI   05H       ; R-STATUS ?
*029F C2 F1 02  018A >        JNZ   FUNP15    ; NEIN
                    018B >; REGISTERANZEIGE
*02A2 3A EC 08  018C >        LDA   AUS+1     ; REGISTERART PRUEFEN
*02A5 CD 70 01  018D >        CALL  RESU      ; ALTES REGISTER SUCHEN
*02A8 09        018E >        DAD   B         ;
*02A9 09        018F >        DAD   B         ; NEUES REGISTER
*02AA 09        0190 >        DAD   B         ; BERECHNEN
*02AB 09        0191 >        DAD   B         ;
*02AC 7E        0192 >        MOV   A,M       ; NEUE REGISTERART
*02AD 23        0193 >        INX   H         ;
*02AE 32 EC 08  0194 >        STA   AUS+1     ;
*02B1 7E        0195 >        MOV   A,M       ;
*02B2 23        0196 >        INX   H         ;
*02B3 32 ED 08  0197 >        STA   AUS+2     ;
*02B6 FE 05     0198 >        CPI   05H       ; r = AKKUMULATOR ?
*02B8 CA 5F 02  0199 >        JZ    FUNPC1    ; JA
*02BB FE 86     019A >FUNP9:  CPI   086H      ; I-REGISTER
*02BD C2 D9 02  019B >        JNZ   FUNP13    ; NEIN
*02C0 3A DE 08  019C >        LDA   IREG      ; INTERRUPT-REGISTER
*02C3 06 07     019D >        MVI   B,7       ; ZAEHLER
*02C5 21 F3 08  019E >FUNP10: LXI   H,AUS+8   ; AUSGABE BITWEISE
*02C8 0E 7E     019F >FUNP11: MVI   C,7EH     ;  0
*02CA 0F        01A0 >        RRC             ; 1 BIT RECHTS SCHIEBEN NACH CARR
*02CB D2 D0 02  01A1 >        JNC   FUNP12    ; CARRY = 0
*02CE 0E 30     01A2 >        MVI   C,30H     ;  1
*02D0 71        01A3 >FUNP12: MOV   M,C       ; BIT NACH AUSGABE
*02D1 2B        01A4 >        DCX   H         ;
*02D2 05        01A5 >        DCR   B         ;
*02D3 C2 C8 02  01A6 >        JNZ   FUNP11    ; SCHLEIFE
*02D6 C3 39 01  01A7 >        JMP   WARTE     ; ANZEIGEN UND WARTEN
```

```
*02D9 FE 01     01A8 >FUNP13: CPI   01H       ; MARKE FUER BEDINGUNGSREGISTER
*02DB C2 E6 02  01A9 >        JNZ   FUNP14    ; NEIN
*02DE 3A D6 08  01AA >        LDA   PSW       ; PSW LADEN
*02E1 06 08     01AB >        MVI   B,8       ; ZAEHLER
*02E3 C3 C5 02  01AC >        JMP   FUNP10    ; BITWEISE AUSGEBEN
*02E6 5E        01AD >FUNP14: MOV   E,M       ; ADRESSE LOW
*02E7 23        01AE >        INX   H         ;
*02E8 56        01AF >        MOV   D,M       ; ADRESSE HIGH
*02E9 1A        01B0 >        LDAX  D         ; ADRESSE LOW
*02EA 13        01B1 >        INX   D         ;
*02EB 6F        01B2 >        MOV   L,A       ;
*02EC 1A        01B3 >        LDAX  D         ; ADRESSE HIGH
*02ED 67        01B4 >        MOV   H,A       ;
*02EE C3 F5 01  01B5 >        JMP   TAST7     ; ANZEIGEN UND WARTEN
*02F1 B7        01B6 >FUNP15: ORA   A         ; .X TESTEN
*02F2 F2 0B 03  01B7 >        JP    FUNP21    ; NEIN  P H E S
                01B8 >; HALTEPUNKTANZEIGE
*02F5 CD 60 01  01B9 >        CALL  HPSU      ; ALTEN HALTEPUNKT SUCHEN
*02F8 09        01BA >        DAD   B         ;
*02F9 09        01BB >        DAD   B         ; NEUEN HALTEPUNKT BERECHNEN
*02FA 09        01BC >        DAD   B         ;
*02FB 7E        01BD >        MOV   A,M       ; NEUEN STATUS
*02FC 23        01BE >        INX   H         ;
*02FD 32 EB 08  01BF >        STA   AUS       ; ANZEIGEN
*0300 5E        01C0 >        MOV   E,M       ; ADRESSE LOW
*0301 23        01C1 >        INX   H         ;
*0302 56        01C2 >        MOV   D,M       ; ADRESSE HIGH
*0303 1A        01C3 >        LDAX  D         ; ADRESSE LOW
*0304 13        01C4 >        INX   D         ;
*0305 6F        01C5 >        MOV   L,A       ;
*0306 1A        01C6 >        LDAX  D         ; ADRESSE HIGH
*0307 67        01C7 >        MOV   H,A       ;
*0308 C3 04 01  01C8 >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
                01C9 >; P  ODER  H  ODER  E  ODER  S  BEFEHLSZAEHLER AENDERN
*030B 2A D4 08  01CA >FUNP21: LHLD  PCR       ; BEFEHLSZAEHLER
*030E 09        01CB >        DAD   B         ;
*030F 22 D4 08  01CC >        SHLD  PCR       ; NEUER WERT
*0312 C3 04 01  01CD >        JMP   AUSHL     ; ANZEIGEN UND WARTEN
                01CE >;
                01CF >; PROGRAMMSTART- UND INTERRUPTEINSPRUENGE
*0315           01D0 >FUNET   EQU   0320H     ; ANSCHLUSS EINZELTAKT
*0315           01D1 >FUNES   EQU   032CH     ; ANSCHLUSS EINZELBEFEHL
*0315           01D2 >FUNGO   EQU   033BH     ; ANSCHLUSS PROGRAMMSTART
                01D3 >;
```

```
                01D4 >; MONITOR - KONSTANTEN - BEREICH
#07A0           01D5 >          ORG   07A0H      ; TABELLEN  E P R O M
                01D6 >TEXT:     DB    00H,01H,01H,7FH,7EH,7FH,5BH,01H,01H ; 8085
07A0 00 01 01 7F 7E 7F 5B 01 01
                01D7 >COTAB:    DB    7EH,30H,6DH,79H,33H,5BH,5FH,72H
07A9 7E 30 6D 79 33 5B 5F 72
                01D8 >          DB    7FH,7BH,77H,1FH,4EH,3DH,4FH,47H
07B1 7F 7B 77 1F 4E 3D 4F 47
*07B9 7E 02     01D9 >FUTAB:    DW    FUNPL      ; + TASTE
*07BB 78 02     01DA >          DW    FUNMI      ; - TASTE
*07BD 2C 03     01DB >          DW    FUNES      ; ES TASTE AEND. 8.1.83
*07BF 4C 02     01DC >          DW    FUNPC      ; P/R TASTE AEND. 8.1.83
*07C1 3B 03     01DD >          DW    FUNGO      ; G-TASTE
*07C3 6D 02     01DE >          DW    FUNHP      ; H TASTE
*07C5 47 02     01DF >          DW    FUNDA      ; D TASTE
*07C7 3C 02     01E0 >          DW    FUNAD      ; A TASTE
*07C9 01 01     01E1 >SYRE:     DB    01H,01H    ; - -
*07CB D6 08     01E2 >          DW    PSW        ; BEDINGUNGSREGISTER
*07CD 77 05     01E3 >          DB    77H,05H    ; A r
*07CF D7 08     01E4 >          DW    AKKU       ; AKKUMULATOR
*07D1 1F 4E     01E5 >          DB    1FH,4EH    ; b C
*07D3 D8 08     01E6 >          DW    BCRE       ; BC-REGISTERPAAR
*07D5 3D 4F     01E7 >          DB    3DH,4FH    ; d E
*07D7 DA 08     01E8 >          DW    DERE       ; DE-REGISTERPAAR
*07D9 37 0E     01E9 >          DB    37H,0EH    ; H L
*07DB DC 08     01EA >          DW    HLRE       ; HL-REGISTERPAAR
*07DD 5B 67     01EB >          DB    5BH,67H    ; S P
*07DF D2 08     01EC >          DW    BSP        ; STAPELZEIGER
*07E1 86 86     01ED >          DB    086H,086H  ; .I .I
*07E3 DE 08     01EE >          DW    IREG       ; INTERRUPTREGISTER
*07E5 01 01     01EF >          DB    01H,01H    ; - -
*07E7 D6 08     01F0 >          DW    PSW        ; BEDINGUNGSREGISTER
*07E9 77 05     01F1 >          DB    77H,05H    ; A r
*07EB D7 08     01F2 >          DW    AKKU       ; AKKUMULATOR
*07ED F9        01F3 >SYHP:     DB    0F9H       ; .3
*07EE E8 08     01F4 >          DW    HPUA+9     ;
*07F0 FE        01F5 >          DB    0FEH       ; .0
*07F1 DF 08     01F6 >          DW    HPUA       ;
*07F3 B0        01F7 >          DB    0B0H       ; .1
*07F4 E2 08     01F8 >          DW    HPUA+3     ;
*07F6 ED        01F9 >          DB    0EDH       ; .2
*07F7 E5 08     01FA >          DW    HPUA+6     ;
*07F9 F9        01FB >          DB    0F9H       ; .3
*07FA E8 08     01FC >          DW    HPUA+9     ;
*07FC FE        01FD >          DB    0FEH       ; .0
*07FD DF 08     01FE >          DW    HPUA       ;
```

```
                01FF >;
                0200 >; MONITOR - RAM - BEREICH
                0201 >; INTERRUPT - PROGRAMM - SPRUENGE
#0800           0202 >          ORG   0800H     ;
*0800           0203 >RST0:     DS    8         ; RST 0
*0808           0204 >RST1:     DS    8         ; RST 1
*0810           0205 >RST2:     DS    8         ; RST 2
*0818           0206 >RST3:     DS    8         ; RST 3
*0820           0207 >RST4:     DS    4         ; RST 4
*0824           0208 >TRAP:     DS    4         ; TRAP-INTERRUPT
*0828           0209 >RST5:     DS    4         ; RST 5
*082C           020A >RST55:    DS    4         ; RST 5.5 INTERRUPT
*0830           020B >RST6:     DS    4         ; RST 6
*0834           020C >RST65:    DS    4         ; RST 6.5 INTERRUPT
*0838           020D >RST7:     DS    4         ; RST 7 = HALTEPUNKT
*083C           020E >RST75:    DS    4         ; RST 7.5 INTERRUPT
*0840           020F >IEND      EQU   0840H     ; ENDE INTERRUPTPROGRAMME
                0210 >;
*0840           0211 >URAM      EQU   1000H     ; ANFANG BENUTZER-RAM
                0212 >;
*0840           0213 >BSTA      EQU   08A0H     ; BENUTZER - STAPEL
                0214 >;
*0840           0215 >MSTA      EQU   08D0H     ; MONITOR - STAPEL
                0216 >;
                0217 >; MONITOR - ARBEITSBEREICH
#08D0           0218 >          ORG   08D0H     ;
*08D0           0219 >LADD:     DS    2         ; LAUFENDE EINGABEADRESSE
*08D2           021A >BSP:      DS    2         ; BENUTZER-STAPELZEIGER
*08D4           021B >PCR:      DS    2         ; BENUTZER-BEFEHLSZAEHLER
*08D6           021C >PSW:      DS    1         ; BENUTZER-BEDINGUNGSREGISTER
*08D7           021D >AKKU:     DS    1         ; BENUTZER-AKKUMULATOR
*08D8           021E >BCRE:     DS    2         ; BENUTZER-BC-REGISTERPAAR
*08DA           021F >DERE:     DS    2         ; BENUTZER-DE-REGISTERPAAR
*08DC           0220 >HLRE:     DS    2         ; BENUTZER-HL-REGISTERPAAR
*08DE           0221 >IREG:     DS    1         ; BENUTZER-INTERRUPT-REGISTER
*08DF           0222 >HPUA:     DS    12        ; 4 HALTEPUNKTE  ADRESSE CODE
*08EB           0223 >AUS:      DS    9         ; AUSGABEPUFFER
*08F4           0224 >MARK1:    DS    1         ; MARKE 1
*08F5           0225 >MARK2:    DS    1         ; MARKE 2
*08F6           0226 >RES:      DS    4         ; RESERVE
*08FA           0227 >ANF:      DS    2         ; ANFANGSADRESSE
*08FC           0228 >END:      DS    2         ; ENDADRESSE
*08FE           0229 >ZIEL:     DS    2         ; ZIELADRESSE
*0900           022A >REND      EQU   0900H     ; RAM - ENDE - MARKE
E0000           022B >          END
```

```
                    0001 >; PROGRAMM STARTEN
                    0002 >; VERSION 24.6.82
*8000               0003 >BSP     EQU   08D2H     ; BENUTZER-STAPELZEIGER
*8000               0004 >PCR     EQU   08D4H     ; BEFEHLSZAEHLER
*8000               0005 >PSW     EQU   08D6H     ; BEDINGUNGSREGISTER
*8000               0006 >AKKU    EQU   08D7H     ; AKKUMULATOR
*8000               0007 >BCRE    EQU   08D8H     ; BR-REGISTERPAAR
*8000               0008 >DERE    EQU   08DAH     ; DE-REGISTERPAAR
*8000               0009 >HLRE    EQU   08DCH     ; HL-REGISTERPAAR
*8000               000A >IREG    EQU   08DEH     ; INTERRUPTREGISTER
*8000               000B >HPUA    EQU   08DFH     ; VIER HALTEPUNKTE ADRESSE CODE
*8000               000C >AUS     EQU   08EBH     ; AUSGABEBEREICH
*8000               000D >AUSHL   EQU   0104H     ; HL ANZEIGEN
*8000               000E >WARTE   EQU   0139H     ; WARTEN AUF TASTENFREIGAGE
*8000               000F >RST7    EQU   0838H     ; HALTEPUNKT-EINSPRUNG
*8000               0010 >TRAP    EQU   0824H     ; TRAP-EINSPRUNG
*8000               0011 >RST75   EQU   083CH     ; RST7.5 EINZELSCHRITT-EINSPRUNG
*8000               0012 >SREG    EQU   08H       ; STEUER/STATUSREGISTER
*8000               0013 >APORT   EQU   09H       ; A-REGISTER
*8000               0014 >BPORT   EQU   0AH       ; B-REGISTER
*8000               0015 >TLOW    EQU   0CH       ; TIMER LOW
*8000               0016 >THIGH   EQU   0DH       ; TIMER HIGH
*8000               0017 >MON5    EQU   0095H     ; MONITOR-VERARBEITUNGSSCHLEIFE
*8000               0018 >MSTA    EQU   08D0H     ; MONITOR-STAPELZEIGER
                    0019 >;
#0320               001A >        ORG   0320H     ; E P R O M
                    001B >; FUNET = EINZELTAKT
*0320 3E 00         001C >FUNET:  MVI   A,00H     ; STATUS LEER
*0322 32 EB 08      001D >        STA   AUS       ;
*0325 D3 09         001E >        OUT   APORT     ; A7 = 0
*0327 D3 0A         001F >        OUT   BPORT     ; STATUS AUS
*0329 C3 83 03      0020 >        JMP   FUNG6     ; REGISTER LADEN TIMER STARTEN
                    0021 >; FUNES = EINZELSCHRITT
*032C CD 39 01      0022 >FUNES:  CALL  WARTE     ; WARTEN AUF TASTENFREIGABE
*032F 3E 4F         0023 >        MVI   A,4FH     ; E
*0331 32 EB 08      0024 >        STA   AUS       ;
*0334 3E 80         0025 >        MVI   A,80H     ; A7=1
*0336 D3 09         0026 >        OUT   APORT     ; EINZELSCHRITT
*0338 C3 83 03      0027 >        JMP   FUNG6     ; REGISTER LADEN TIMER STARTEN
                    0028 >; FUNGO = PROGRAMM STARTEN
*033B CD 39 01      0029 >FUNGO:  CALL  WARTE     ; WARTEN AUF TASTENFREIGABE
*033E 3E 01         002A >        MVI   A,01H     ; - NACH STATUS
*0340 32 EB 08      002B >        STA   AUS       ;
*0343 D3 0A         002C >        OUT   BPORT     ; AUSGEBEN
*0345 3E 00         002D >        MVI   A,00H     ; LINKE STELLE
*0347 D3 09         002E >        OUT   APORT     ;
                    002F >; HALTEPUNKTE LADEN
*0349 06 04         0030 >        MVI   B,4       ; ZAEHLER
*034B 21 DF 08      0031 >        LXI   H,HPUA    ; ANFANG HALTEPUNKTLISTE
*034E 5E            0032 >FUNG1:  MOV   E,M       ; ADRESSE LOW
*034F 23            0033 >        INX   H         ;
*0350 56            0034 >        MOV   D,M
*0351 23            0035 >        INX   H         ;
*0352 7A            0036 >        MOV   A,D       ;
*0353 B3            0037 >        ORA   E         ; ADRESSE = 0 KEIN HALTEPUNKT ?
*0354 C2 5F 03      0038 >        JNZ   FUNG3     ; HALTEPUNKT GESETZT
```

```
*0357 23        0039 >FUNG2:  INX   H         ; KEIN HALTEPUNKT
*0358 05        003A >        DCR   B         ;
*0359 C2 4E 03  003B >        JNZ   FUNG1     ; NAECHSTEN HALTEPUNKT
*035C C3 7D 03  003C >        JMP   FUNG5     ;
*035F 1A        003D >FUNG3:  LDAX  D         ; CODE LADEN
*0360 77        003E >        MOV   M,A       ; RETTEN
*0361 3A D4 08  003F >        LDA   PCR       ; STARTADRESSE LOW
*0364 BB        0040 >        CMP   E         ; = HALTEPUNKTADRESSE LOW ?
*0365 C2 6F 03  0041 >        JNZ   FUNG4     ; UNGLEICH
*0368 3A D5 08  0042 >        LDA   PCR+1     ; STARTADRESSE HIGH
*036B BA        0043 >        CMP   D         ; = HALTEPUNKTADRESSE HIGH ?
*036C CA 57 03  0044 >        JZ    FUNG2     ; JA: HALTEPUNKT NICHT SETZEN
*036F 3E FF     0045 >FUNG4:  MVI   A,0FFH    ; RST7-CODE = HALTEPUNKT
*0371 12        0046 >        STAX  D         ; ERSETZT ALTEN CODE
*0372 E5        0047 >        PUSH  H         ; HL RETTEN
*0373 21 BA 03  0048 >        LXI   H,INHAP   ; EINSPRUNG RST 7
*0376 22 3A 08  0049 >        SHLD  RST7+2    ;
*0379 E1        004A >        POP   H         ; HL ZURUECK
*037A C3 57 03  004B >        JMP   FUNG2     ;
*037D 21 21 04  004C >FUNG5:  LXI   H,INTRAP  ; EINSPRUNGPUNKT TRAP
*0380 22 26 08  004D >        SHLD  TRAP+2    ;
                004E >; EINSPRUNGPUNKT FUER EINZELSCHRITT  REGISTER RETTEN
*0383 2A D2 08  004F >FUNG6:  LHLD  BSP       ; BENUTZER-STAPELZEIGER
*0386 F9        0050 >        SPHL            ;
*0387 2A D4 08  0051 >        LHLD  PCR       ; STARTADRESSE
*038A E5        0052 >        PUSH  H         ; NACH STAPEL
*038B 2A D8 08  0053 >        LHLD  BCRE      ; BC-REGISTERPAAR
*038E 44        0054 >        MOV   B,H       ;
*038F 4D        0055 >        MOV   C,L       ;
*0390 2A DA 08  0056 >        LHLD  DERE      ; DE-REGISTERPAAR
*0393 EB        0057 >        XCHG            ;
*0394 3A EB 08  0058 >        LDA   AUS       ; STATUS ?
*0397 FE 01     0059 >        CPI   01H       ; EINZELSCHRITT ?
*0399 CA AD 03  005A >        JZ    FUNG7     ; NEIN
                005B >; EINZELSCHRITT  TIMER STARTEN
*039C 21 1B 04  005C >        LXI   H,INRS75  ;
*039F 22 26 08  005D >        SHLD  TRAP+2    ; ODER   SHLD  RST75+2
*03A2 3E 8E     005E >        MVI   A,142     ; TIMERWERT 72 ZYKLEN
*03A4 D3 0C     005F >        OUT   TLOW      ; TIMER LOW
*03A6 AF        0060 >        XRA   A         ; NULL
*03A7 D3 0D     0061 >        OUT   THIGH     ; TIMER HIGH
*03A9 3E C3     0062 >        MVI   A,0C3H    ; TIMER STARTEN
*03AB D3 08     0063 >        OUT   SREG      ;
*03AD 2A D6 08  0064 >FUNG7:  LHLD  PSW       ; AKKU UND PSW
*03B0 E5        0065 >        PUSH  H         ; AKKU UND PSW
*03B1 F1        0066 >        POP   PSW       ;
*03B2 2A DC 08  0067 >        LHLD  HLRE      ; HL-REGISTERPAAR
*03B5 00        0068 >        NOP
*03B6 00        0069 >        NOP
*03B7 C9        006A >        RET             ; PROGRAMM STARTEN
*03B8 00        006B >        NOP
*03B9 00        006C >        NOP
```

```
                    006D >; EINSPRUNG HALTEPUNKT
*03BA 32 D7 08  006E >INHAP:  STA   AKKU      ; AKKU RETTEN
*03BD 3E 37     006F >        MVI   A,37H     ; H
*03BF 32 EB 08  0070 >        STA   AUS       ; STATUS
                0071 >; REGISTER RETTEN FUER ALLE EINSPRUENGE S H UND E
*03C2 22 DC 08  0072 >INHAP1: SHLD  HLRE      ; HL-REGISTERPAAR
*03C5 EB        0073 >        XCHG            ;
*03C6 22 DA 08  0074 >        SHLD  DERE      ; DE-REGISTERPAAR
*03C9 60        0075 >        MOV   H,B       ;
*03CA 69        0076 >        MOV   L,C       ;
*03CB 22 D8 08  0077 >        SHLD  BCRE      ; BC-REGISTERPAAR
*03CE 20        0078 >        RIM             ;
*03CF 32 DE 08  0079 >        STA   IREG      ; INTERRUPTREGISTER
*03D2 E1        007A >        POP   H         ;
*03D3 22 D4 08  007B >        SHLD  PCR       ; BEFEHLSZAEHLER
*03D6 F5        007C >        PUSH  PSW       ;
*03D7 E1        007D >        POP   H         ;
*03D8 7D        007E >        MOV   A,L       ;
*03D9 32 D6 08  007F >        STA   PSW       ; BEDINGUNGSREGISTER
*03DC 21 00 00  0080 >        LXI   H,0000H   ;
*03DF 39        0081 >        DAD   SP        ;
*03E0 22 D2 08  0082 >        SHLD  BSP       ; BENUTZER-STAPELZEIGER
*03E3 31 D0 08  0083 >        LXI   SP,MSTA   ; SYSTEM-STAPELZEIGER
*03E6 3A EB 08  0084 >        LDA   AUS       ; STATUS ?
*03E9 FE 4F     0085 >        CPI   4FH       ; E ?
*03EB CA 2C 04  0086 >        JZ    MONI      ; PCR ANZEIGEN
                0087 >; HALTEPUNKTE RETTEN FUER  S  UND  H
*03EE 06 04     0088 >        MVI   B,4       ; ZAEHLER
*03F0 21 DF 08  0089 >        LXI   H,HPUA    ; ANFANGSADRESSE HALTEPUNKTLISTE
*03F3 5E        008A >INHAP2: MOV   E,M       ; ADRESSE LOW
*03F4 23        008B >        INX   H         ;
*03F5 56        008C >        MOV   D,M       ; ADRESSE HIGH
*03F6 23        008D >        INX   H         ;
*03F7 7A        008E >        MOV   A,D       ;
*03F8 B3        008F >        ORA   E         ; HALTEPUNKT GESETZT ?
*03F9 C2 04 04  0090 >        JNZ   INHAP4    ; JA: CODE ZURUECKLADEN
*03FC 23        0091 >INHAP3: INX   H         ; NICHT GESETZT
*03FD 05        0092 >        DCR   B         ; ZAEHLER - 1
*03FE C2 F3 03  0093 >        JNZ   INHAP2    ;
*0401 C3 09 04  0094 >        JMP   INHAP5    ;
*0404 7E        0095 >INHAP4: MOV   A,M       ; ALTEN CODE
*0405 12        0096 >        STAX  D         ; ZURUECKSCHREIBEN
*0406 C3 FC 03  0097 >        JMP   INHAP3    ;
*0409 3A EB 08  0098 >INHAP5: LDA   AUS       ; STATUS ?
*040C FE 5B     0099 >        CPI   5BH       ; S ?
*040E CA 2C 04  009A >        JZ    MONI      ; PCR ANZEIGEN
                009B >; BEI HALTEPUNKT PC = PC - 1
*0411 2A D4 08  009C >        LHLD  PCR       ; BEFEHLSZAEHLER
*0414 2B        009D >        DCX   H         ; -1
*0415 22 D4 08  009E >        SHLD  PCR       ;
*0418 C3 2C 04  009F >        JMP   MONI      ; PCR ANZEIGEN
                00A0 >; EINSPRUNG EINZELSCHRITT
*041B 32 D7 08  00A1 >INRS75: STA   AKKU      ; AKKU RETTEN
*041E C3 C2 03  00A2 >        JMP   INHAP1    ; REGISTER RETTEN
                00A3 >; EINSPRUNG  STOP-TASTE
*0421 32 D7 08  00A4 >INTRAP: STA   AKKU      ; AKKU RETTEN
*0424 3E 5B     00A5 >        MVI   A,5BH     ; S
*0426 32 EB 08  00A6 >        STA   AUS       ; STATUS
*0429 C3 C2 03  00A7 >        JMP   INHAP1    ; REGISTER UND HALTEPUNKTE RETTEN
*042C 2A D4 08  00A8 >MONI:   LHLD  PCR       ; BENUTZER-BEFEHLSZAEHLER
*042F CD 04 01  00A9 >        CALL  AUSHL     ; ANZEIGEN
*0432 C3 95 00  00AA >        JMP   MON5      ; VERARBEITUNGSSCHLEIFE
E0000           00AB >        END
```

```
                        0001 >; HILFSFUNKTIONEN  H I L F  VERSION 7.1.83
*8000                   0002 >PCR      EQU   08D4H     ; BEFEHLSZAEHLER
*8000                   0003 >PSW      EQU   08D6H     ; BEDINGUNGSREGISTER UND AKKU
*8000                   0004 >AKKU     EQU   08D7H     ; AKKU ALLEIN
*8000                   0005 >BCRE     EQU   08D8H     ; BC-REGISTERPAAR
*8000                   0006 >DERE     EQU   08DAH     ; DE-REGISTERPAAR
*8000                   0007 >HLRE     EQU   08DCH     ; HL-REGISTERPAAR
*8000                   0008 >MONW     EQU   0069H     ; MONITOR-WARM-START
*8000                   0009 >MON5     EQU   0095H     ; VERARBEITUNGSSCHLEIFE
*8000                   000A >AUSHL    EQU   0104H     ; HL UND BYTE (M) ANZEIGEN
*8000                   000B >AUS      EQU   08EBH     ; AUSGABEBEREICH
*8000                   000C >ANZ      EQU   00BAH     ; ANZEIGEUNTERPROGRAMM
*8000                   000D >MS1      EQU   00ABH     ; N * 1 MS WARTEN
*8000                   000E >ABUSL    EQU   00        ; ADRESSLEITUNGEN LOW
*8000                   000F >DBUS     EQU   01        ; DATENLEITUNGEN
*8000                   0010 >ABUSH    EQU   02        ; VPP VCC FREI OE CE A10 A9 A8
*8000                   0011 >SBUS     EQU   03        ; STEUERREGISTER 8255
#0440                   0012 >         ORG   0440H     ; ADRESSZAEHLER
LF440                   0013 >         LOAD  0F440H    ; LADEZAEHLER
*0440 C3 BF 04          0014 >HILFE    JMP   HILF      ; NACH HAUPTPROGAMM
                        0015 >; FEHLERPROGRAMME UND UNTERPROGRAMME
                        0016 >; CODE  =  UNBEKANNTER CODE IM AKKU
*0443 21 47 0D          0017 >CODE     LXI   H,0D47H   ; c F
*0446 22 EB 08          0018 >         SHLD  AUS       ;
*0449 21 1D 3D          0019 >         LXI   H,3D1DH   ; d o
*044C 22 ED 08          001A >         SHLD  AUS+2     ;
*044F C3 95 00          001B >         JMP   MON5      ; ANZEIGESCHLEIFE
                        001C >; ANKEN = ANFANGSADRESSE GROESSER ENDADRESSE
*0452 21 47 77          001D >ANKEN    LXI   H,7747H   ; A F
*0455 22 EB 08          001E >         SHLD  AUS       ;
*0458 21 01 4F          001F >         LXI   H,4F01H   ; E -
*045B 22 ED 08          0020 >         SHLD  AUS+2     ;
*045E C3 95 00          0021 >         JMP   MON5      ; ANZEIGESCHLEIFE
                        0022 >; ENDE = NORMALES FUNKTIONSENDE
*0461 21 4F 15          0023 >ENDE     LXI   H,154FH   ; n E
*0464 22 EC 08          0024 >         SHLD  AUS+1     ;
*0467 21 3D 4F          0025 >         LXI   H,4F3DH   ; E d
*046A 22 EE 08          0026 >         SHLD  AUS+3     ;
*046D C3 95 00          0027 >         JMP   MON5      ; ANZEIGESCHLEIFE
                        0028 >; FEHLG = FEHLERMELDUNG MIT FORTSETZUNGSMOEGLICHKEIT
*0470 22 DC 08          0029 >FEHLG    SHLD  HLRE      ; HL RETTEN
*0473 EB                002A >         XCHG            ;
*0474 22 DA 08          002B >         SHLD  DERE      ; DE RETTEN
*0477 F5                002C >         PUSH  PSW       ;
*0478 E1                002D >         POP   H         ;
*0479 22 D6 08          002E >         SHLD  PSW       ; PSW UND AKKU RETTEN
*047C E1                002F >         POP   H         ; BEFEHLSZAEHLER RETTEN
*047D 22 D4 08          0030 >         SHLD  PCR       ; FUER FORTSETZUNG MIT GO
*0480 C5                0031 >         PUSH  B         ;
*0481 E1                0032 >         POP   H         ;
*0482 22 D8 08          0033 >         SHLD  BCRE      ; BC RETTEN
*0485 2A DC 08          0034 >         LHLD  HLRE      ; HL ENTHAELT FEHLERADRESSE
*0488 3E 47             0035 >         MVI   A,47H     ; F ALS FEHLERMARKE
*048A 32 EB 08          0036 >         STA   AUS       ; ANZEIGEN
*048D CD 04 01          0037 >         CALL  AUSHL     ; FEHLERADRESSE UND BYTE ANZEIGEN
*0490 CD BA 00          0038 >         CALL  ANZ       ; ANZEIGEZYKLUS ZUM ENTPRELLEN
*0493 C3 95 00          0039 >         JMP   MON5      ; ANZEIGESCHLEIFE
```

```
                    003A >; BCVDE = 16-BIT-VERGLEICH  BC - DE
                    003B >; NULL: BEIDE GLEIC  C=1: BC KLEINER DE   C=0: BC GROESSE
*0496 E5            003C >BCVDE   PUSH  H          ; HL RETTEN
*0497 67            003D >        MOV   H,A        ; AKKU RETTEN
*0498 78            003E >        MOV   A,B        ; B LADEN
*0499 BA            003F >        CMP   D          ; MIT D VERGLEICHEN
*049A C2 9F 04      0040 >        JNZ   BCVDE1     ; UNGLEICH: FERTIG
*049D 79            0041 >        MOV   A,C        ; C LADEN
*049E BB            0042 >        CMP   E          ; MIT E VERGLEICHEN
*049F 7C            0043 >BCVDE1  MOV   A,H        ; A ZURUECK
*04A0 E1            0044 >        POP   H          ; HL ZURUECK
*04A1 C9            0045 >        RET              ;
                    0046 >; TIME = WARTEN HALBE BITZEIT  ZAEHLER IN E
                    0047 >;       CALL  TIME       ; 18 TAKTE
*04A2 D5            0048 >TIME    PUSH  D          ; 12 TAKTE
*04A3 1D            0049 >TIME1   DCR   E          ; 4  TAKTE
*04A4 C2 A3 04      004A >        JNZ   TIME1      ; 10 TAKTE; BEI NEIN 7 TAKTE
*04A7 D1            004B >        POP   D          ; 10 TAKTE
*04A8 C9            004C >        RET              ; 10 TAKTE
                    004D >; DEIN = EPROM LESEN VCC=VPP=5 V ADRESSE IN BC
*04A9 79            004E >DEIN    MOV   A,C        ; ADRESSE L
*04AA D3 00         004F >        OUT   ABUSL      ; AUSGEBEN
*04AC 78            0050 >        MOV   A,B        ; ADRESSE H
*04AD E6 07         0051 >        ANI   07H        ; 000 00 111
*04AF F6 80         0052 >        ORI   80H        ; 100 00 000
*04B1 D3 02         0053 >        OUT   ABUSH      ; OE = CE = LOW
*04B3 DB 01         0054 >        IN    DBUS       ; DATEN LESEN
*04B5 F5            0055 >        PUSH  PSW        ; RETTEN
*04B6 78            0056 >        MOV   A,B        ; ADRESSE H
*04B7 E6 07         0057 >        ANI   07H        ; 000 00 111
*04B9 F6 98         0058 >        ORI   98H        ; 100 11 000
*04BB D3 02         0059 >        OUT   ABUSH      ; OE = CE = HIGH
*04BD F1            005A >        POP   PSW        ; DATEN ZURUECK
*04BE C9            005B >        RET              ;
                    005C >; HAUPTPROGRAMM HILFSFUNKTIONEN
*04BF E5            005D >HILF    PUSH  H          ; HL RETTEN
*04C0 21 00 00      005E >        LXI   H,0        ; LOESCHEN
*04C3 22 EE 08      005F >        SHLD  AUS+3      ;
*04C6 22 F0 08      0060 >        SHLD  AUS+5      ;
*04C9 22 F2 08      0061 >        SHLD  AUS+7      ;
*04CC E1            0062 >        POP   H          ; HL ZURUECK
*04CD FE 00         0063 >        CPI   00H        ; CODE = 00
*04CF CA 43 04      0064 >        JZ    CODE       ; FEHLER CODE 00 ALS FALLE
*04D2 FE 01         0065 >        CPI   01H        ;
*04D4 C2 F1 04      0066 >        JNZ   HILF2      ; NEIN
                    0067 >; CODE = OFILL BC=ANFA  DE=ENDA  L=BYTE
*04D7 13            0068 >        INX   D          ; ENDADRESSE +1
*04D8 CD 96 04      0069 >        CALL  BCVDE      ; VERGLEICHEN
*04DB D2 52 04      006A >        JNC   ANKEN      ; FEHLER: ANFA GROESSER ENDA
*04DE 7D            006B >        MOV   A,L        ; ZEICHEN LADEN
*04DF C5            006C >        PUSH  B          ; BC AUCH NACH HL
*04E0 E1            006D >        POP   H          ;
*04E1 77            006E >HILF1   MOV   M,A        ; SPEICHERN
*04E2 BE            006F >        CMP   M          ; VERGLEICHEN
*04E3 C4 70 04      0070 >        CNZ   FEHLG      ; UNGLEICH: FEHLER
*04E6 03            0071 >        INX   B          ; ADRESSE + 1
*04E7 23            0072 >        INX   H          ; ADRESSE + 1
*04E8 CD 96 04      0073 >        CALL  BCVDE      ; ENDE ?
*04EB DA E1 04      0074 >        JC    HILF1      ; NEIN: WEITER
*04EE C3 61 04      0075 >        JMP   ENDE       ; JA: FERTIG
*04F1 FE 02         0076 >HILF2   CPI   02H        ; CODE 02 ?
*04F3 C2 39 05      0077 >        JNZ   HILF6      ; NEIN
```

```
                    0078 >; CODE = 02: MOVE  BC=ANFA  DE=ENDE  HL=ZIEL
*04F6 13            0079 >         INX   D          ; ENDADRESSE + 1
*04F7 CD 96 04      007A >         CALL  BCVDE      ; VERGLEICHEN
*04FA D2 52 04      007B >         JNC   ANKEN      ; FEHLER: ANFA GROESSER ENDA
*04FD EB            007C >         XCHG             ; HL=ENDA DE=ZIEL
*04FE CD 96 04      007D >         CALL  BCVDE      ; AUFWAERTS ODER ABWAERTS?
*0501 CA 61 04      007E >         JZ    ENDE       ; ANFA = ZIEL
*0504 DA 19 05      007F >         JC    HILF4      ; ANFA KLEINER ZIEL
*0507 EB            0080 >         XCHG             ; ANFA GROESSER ZIEL
*0508 0A            0081 >HILF3    LDAX  B          ; WERT LADEN
*0509 77            0082 >         MOV   M,A        ; SPEICHERN
*050A BE            0083 >         CMP   M          ; VERGLEICHEN
*050B C4 70 04      0084 >         CNZ   FEHLG      ; UNGLEICH: FEHLER
*050E 03            0085 >         INX   B          ; HERADRESSE + 1
*050F 23            0086 >         INX   H          ; ZIELADRESSE + 1
*0510 CD 96 04      0087 >         CALL  BCVDE      ; ENDE ?
*0513 DA 08 05      0088 >         JC    HILF3      ; NEIN: WEITER
*0516 C3 61 04      0089 >         JMP   ENDE       ; JA: FERTIG
*0519 E5            008A >HILF4    PUSH  H          ; HERADRESSE + 1 RETTEN
*051A 78            008B >         MOV   A,B        ; BC KOMPLEMENTIEREN
*051B 2F            008C >         CMA              ;
*051C 47            008D >         MOV   B,A        ;
*051D 79            008E >         MOV   A,C        ;
*051E 2F            008F >         CMA              ;
*051F 4F            0090 >         MOV   C,A        ;
*0520 03            0091 >         INX   B          ; BC IM 2ER KOMPLEMENT
*0521 09            0092 >         DAD   B          ; HL = HL - BC = DIFFERENZ
*0522 44            0093 >         MOV   B,H        ; BC = ZAEHLER
*0523 4D            0094 >         MOV   C,L        ;
*0524 19            0095 >         DAD   D          ; HL = ZIEL-ANFANGSADRESSE
*0525 D1            0096 >         POP   D          ; DE = HER-ENDADRESSE
*0526 1B            0097 >         DCX   D          ;
*0527 2B            0098 >         DCX   H          ;
*0528 1A            0099 >HILF5    LDAX  D          ; WERT LADEN
*0529 77            009A >         MOV   M,A        ; WERT SPEICHERN
*052A BE            009B >         CMP   M          ; VERGLEICHEN
*052B C4 70 04      009C >         CNZ   FEHLG      ; UNGLEICH: FEHLER
*052E 2B            009D >         DCX   H          ; ZIELADRESSE - 1
*052F 1B            009E >         DCX   D          ; HERADRESSE - 1
*0530 0B            009F >         DCX   B          ; ZAEHLER - 1
*0531 78            00A0 >         MOV   A,B        ; AUF NULL PRUEFEN
*0532 B1            00A1 >         ORA   C          ;
*0533 C2 28 05      00A2 >         JNZ   HILF5      ; UNGLEICH NULL: WEITER
*0536 C3 61 04      00A3 >         JMP   ENDE       ; GLEICH NULL: FERTIG
*0539 FE 03         00A4 >HILF6    CPI   03H        ; CODE 03 ?
*053B C2 94 05      00A5 >         JNZ   HILF7      ; NEIN
```

```
                    00A6 >; CODE=03 : SERIELLE EINGABE BC=ANF DE=END L=BAUDRATE
*053E 13            00A7 >          INX   D          ; ENDADRESSE + 1
*053F CD 96 04      00A8 >          CALL  BCVDE      ; VERGLEICHEN
*0542 D2 52 04      00A9 >          JNC   ANKEN      ; FEHLER BC GROESSER DE
*0545 1B            00AA >          DCX   D          ; ENDADRESSE - 1
*0546 EB            00AB >          XCHG             ; HL=ENDA  E=BAUDRATE
*0547 E5            00AC >          PUSH  H          ; HL MIT BC VERTAUSCHEN
*0548 C5            00AD >          PUSH  B          ;
*0549 E1            00AE >          POP   H          ;
*054A C1            00AF >          POP   B          ;
*054B CD 66 05      00B0 >EIN       CALL  SEIN       ; BYTE LESEN
*054E DC 70 04      00B1 >          CC    FEHLG      ; FEHLER
*0551 D3 18         00B2 >          OUT   18H        ; KONTROLAUSGABE
*0553 77            00B3 >          MOV   M,A        ; SPEICHERAUSGABE
*0554 BE            00B4 >          CMP   M          ; VERGLEICHEN
*0555 C4 70 04      00B5 >          CNZ   FEHLG      ; UNGLEICH: FEHLER
*0558 78            00B6 >          MOV   A,B        ; ENDE ?
*0559 BC            00B7 >          CMP   H          ;
*055A C2 62 05      00B8 >          JNZ   EIN1       ; UNGLEICH
*055D 79            00B9 >          MOV   A,C        ;
*055E BD            00BA >          CMP   L          ;
*055F CA 61 04      00BB >          JZ    ENDE       ; FERTIG
*0562 23            00BC >EIN1      INX   H          ; ADRESSE + 1
*0563 C3 4B 05      00BD >          JMP   EIN        ; SCHLEIFE
                    00BE >; SEIN = SID-EINGABE 8 BIT 2 STOP OHNE PAR  E=RATE
*0566 20            00BF >SEIN      RIM              ; LEITUNG LESEN
*0567 B7            00C0 >          ORA   A          ; AKKU TESTEN
*0568 F2 92 05      00C1 >          JP    SEINF      ; LEITUNG LOW: FEHLER
*056B E5            00C2 >          PUSH  H          ; HL RETTEN
*056C 26 08         00C3 >          MVI   H,8        ; BITZAEHLER
*056E 2E 00         00C4 >          MVI   L,0        ; DATENBYTE
*0570 20            00C5 >SEIN1     RIM              ; LEITUNG HIGH: RUHEZUSTAND
*0571 B7            00C6 >          ORA   A          ; AKKU TESTEN
*0572 FA 70 05      00C7 >          JM    SEIN1      ; IMMER NOCH HIGH
                    00C8 >; FALLENDE FLANKE STARTBIT ERKANNT
*0575 CD A2 04      00C9 >          CALL  TIME       ; HALBE BITZEIT WARTEN
*0578 20            00CA >          RIM              ; LEITUNG LESEN
*0579 B7            00CB >          ORA   A          ; AKKU TESTEN
*057A FA 92 05      00CC >          JM    SEINF      ; LEITUNG HIGH: KEIN STARTBIT
*057D CD A2 04      00CD >SEIN2     CALL  TIME       ; HALBE BITZEIT WARTEN
*0580 CD A2 04      00CE >          CALL  TIME       ; HALBE BITZEIT WARTEN
*0583 20            00CF >          RIM              ; BIT LESEN
*0584 07            00D0 >          RLC              ; NACH CARRYBIT
*0585 7D            00D1 >          MOV   A,L        ; ALTES BYTE HOLEN
*0586 1F            00D2 >          RAR              ; NEUES BIT EINBAUEN
*0587 6F            00D3 >          MOV   L,A        ; NEUES BYTE RETTEN
*0588 25            00D4 >          DCR   H          ; BITZAEHLER - 1
*0589 C2 7D 05      00D5 >          JNZ   SEIN2      ; UNGLEICH NULL
*058C CD A2 04      00D6 >          CALL  TIME       ; HALBE BITZEIT WARTEN
*058F B7            00D7 >          ORA   A          ; FERTIG: C=0
*0590 E1            00D8 >          POP   H          ; HL ZURUECK
*0591 C9            00D9 >          RET              ;
*0592 37            00DA >SEINF     STC              ; FEHLER: C=1
*0593 C9            00DB >          RET              ;
*0594 FE 04         00DC >HILF7     CPI   04H        ; CODE 04 ?
*0596 C2 DE 05      00DD >          JNZ   HILF8      ; NEIN
```

```
                    00DE >;CODE = 04: AUSGABE SOD BC=ANFA  DE=ENDA  L=RATE
*0599 13            00DF >          INX   D           ; ENDADRESSE + 1
*059A CD 96 04      00E0 >          CALL  BCVDE       ; VERGLEICHEN
*059D D2 52 04      00E1 >          JNC   ANKEN       ; FEHLER
*05A0 1B            00E2 >          DCX   D           ; ENDADRESSE - 1
*05A1 EB            00E3 >          XCHG              ; HL=ENDADR  E = RATE
*05A2 0A            00E4 >AUS1      LDAX  B           ; BYTE HOLEN
*05A3 D3 18         00E5 >          OUT   18H         ; KONTROLLAUSGABE
*05A5 CD B6 05      00E6 >          CALL  SAUS        ; BYTE AUSGEBEN
*05A8 78            00E7 >          MOV   A,B         ; ENDADRESSE LOW
*05A9 BC            00E8 >          CMP   H           ; VERGLEICHEN
*05AA C2 B2 05      00E9 >          JNZ   AUS2        ; UNGLEICH
*05AD 79            00EA >          MOV   A,C         ; ENDADRESSE LOW
*05AE BD            00EB >          CMP   L           ; VERGLEICHEN
*05AF CA 61 04      00EC >          JZ    ENDE        ; FERTIG
*05B2 03            00ED >AUS2      INX   B           ; ADRESSE+ 1
*05B3 C3 A2 05      00EE >          JMP   AUS1        ; SCHLEIFE
                    00EF >; SAUS = AUSGABE SOD 8 BIT  2 STOPBIT OHNE PAR E = RATE
*05B6 C5            00F0 >SAUS      PUSH  B           ; BC RETTEN
*05B7 E5            00F1 >          PUSH  H           ; HL RETTEN
*05B8 06 40         00F2 >          MVI   B,40H       ; MARKE SOD-BIT
*05BA 26 09         00F3 >          MVI   H,9         ; BITZAEHLER + STARTBIT
*05BC 6F            00F4 >          MOV   L,A         ; BYTE RETTEN
*05BD 3E 40         00F5 >          MVI   A,40H       ; STARTBIT LADEN
*05BF 30            00F6 >SAUS1     SIM               ; BIT AUSGEBEN
*05C0 CD A2 04      00F7 >          CALL  TIME        ; HALBE BITZEIT
*05C3 CD A2 04      00F8 >          CALL  TIME        ; HALBE BITZEIT
*05C6 7D            00F9 >          MOV   A,L         ; 4 TAKTE BYTE HOLEN
*05C7 0F            00FA >          RRC               ; 4 TAKTE BIT NACH CARRY
*05C8 6F            00FB >          MOV   L,A         ; 4 TAKTE NEUES BYTE RETTEN
*05C9 1F            00FC >          RAR               ; 4 TAKTE BIT ZURUECK
*05CA B0            00FD >          ORA   B           ; 4 TAKTE SOD-MARKE EINBAUEN
*05CB 25            00FE >          DCR   H           ; 4 TAKTE BITZAEHLER
*05CC C2 BF 05      00FF >          JNZ   SAUS1       ; 10 TAKTE UNGLEICH NULL: AUS
*05CF 3E C0         0100 >          MVI   A,0C0H      ; FERTIG: 11000000
*05D1 30            0101 >          SIM               ; STOP-BITS AUSGEBEN
*05D2 3E 05         0102 >          MVI   A,5         ; ZAEHLER 2 1/2 BITZEITEN
*05D4 CD A2 04      0103 >SAUS2     CALL  TIME        ; HALBE BITZEIT
*05D7 3D            0104 >          DCR   A           ; HERABZAEHLEN
*05D8 C2 D4 05      0105 >          JNZ   SAUS2       ;
*05DB E1            0106 >          POP   H           ; HL ZURUECK
*05DC C1            0107 >          POP   B           ; BC ZURUECK
*05DD C9            0108 >          RET               ; FERTIG
*05DE FE 05         0109 >HILF8     CPI   05H         ; CODE 05 ?
*05E0 C2 0A 06      010A >          JNZ   HILF10      ; NEIN
```

```
                    010B >; CODE=05: EPROM TESTEN BC=ANF DE=END
*05E3 13            010C >          INX   D          ; ENDADRESSE + 1
*05E4 CD 96 04      010D >          CALL  BCVDE      ; VERGLEICHEN
*05E7 D2 52 04      010E >          JNC   ANKEN      ; FEHLER BC GROESSER DE
*05EA 3E 82         010F >          MVI   A,82H      ; A=AUS B=EIN (DATEN) C=AUS
*05EC D3 03         0110 >          OUT   SBUS       ; DATEN AUS EPROM LESEN
*05EE 3E 98         0111 >          MVI   A,98H      ; 100 11 000
*05F0 D3 02         0112 >          OUT   ABUSH      ; VPP=VCC=5 V OE=CE=HIGH
*05F2 CD A9 04      0113 >HILF9     CALL  DEIN       ; BYTE LESEN
*05F5 60            0114 >          MOV   H,B        ; WEGEN FEHLERANZEIGE
*05F6 69            0115 >          MOV   L,C        ;
*05F7 FE FF         0116 >          CPI   0FFH       ; UNPROGRAMMIERT ?
*05F9 C4 70 04      0117 >          CNZ   FEHLG      ; NEIN: FEHLER
*05FC 03            0118 >          INX   B          ; ADRESSE + 1
*05FD CD 96 04      0119 >          CALL  BCVDE      ; VERGLEICHEN ENDE
*0600 DA F2 05      011A >          JC    HILF9      ; NEIN
*0603 3E 9B         011B >          MVI   A,9BH      ; ALLES NULL UND EINGANG
*0605 D3 03         011C >          OUT   SBUS       ; ALLE PINS TRISTATE
*0607 C3 61 04      011D >          JMP   ENDE       ; FERTIG
*060A FE 06         011E >HILF10    CPI   06H        ; CODE 06 ?
*060C C2 51 06      011F >          JNZ   HILF12     ; NEIN
                    0120 >; CODE=06: EPROM PROGRAMMIEREN BC=ANF DE=END HL=RAM
*060F 13            0121 >          INX   D          ; ENDADRESSE + 1
*0610 CD 96 04      0122 >          CALL  BCVDE      ; VERGLEICHEN
*0613 D2 52 04      0123 >          JNC   ANKEN      ; FEHLER: BC GROESSER DE
*0616 3E 80         0124 >          MVI   A,80H      ; A=AUS B=AUS(DATEN) C=AUS
*0618 D3 03         0125 >          OUT   SBUS       ; PROGRAMMIEREN
*061A 3E 00         0126 >          MVI   A,00       ; 000 00 000
*061C D3 02         0127 >          OUT   ABUSH      ; VPP=25V VCC=5V OE=CE=LOW
*061E 79            0128 >HILF11    MOV   A,C        ; ADRESSE L
*061F D3 00         0129 >          OUT   ABUSL      ; AUSGEBEN
*0621 78            012A >          MOV   A,B        ; ADRESSE H
*0622 E6 07         012B >          ANI   07H        ; 000 00 111
*0624 F6 10         012C >          ORI   10H        ; 000 10 000
*0626 D3 02         012D >          OUT   ABUSH      ; OE=HIGH CE=PGM=LOW
*0628 7E            012E >          MOV   A,M        ; DATEN HOLEN
*0629 D3 01         012F >          OUT   DBUS       ; AUSGEBEN
*062B 78            0130 >          MOV   A,B        ; ADRESSE H
*062C E6 07         0131 >          ANI   07H        ; 000 00 111
*062E F6 18         0132 >          ORI   18H        ; 000 11 000
*0630 D3 02         0133 >          OUT   ABUSH      ; OE=HIGH CE=PGM=HIGH
*0632 3E 32         0134 >          MVI   A,50       ; 50 MS
*0634 CD AB 00      0135 >          CALL  MS1        ; WARTEN
*0637 78            0136 >          MOV   A,B        ; ADRESSE H
*0638 E6 07         0137 >          ANI   07H        ; 000 00 111
*063A F6 10         0138 >          ORI   10H        ; 000 10 000
*063C D3 02         0139 >          OUT   ABUSH      ; OE=HIGH CE=PGM=LOW
*063E E6 07         013A >          ANI   07H        ; 000 00 111
*0640 D3 02         013B >          OUT   ABUSH      ; OE=LOW  CE=PGM=LOW
*0642 03            013C >          INX   B          ; EPROM-ADRESSE + 1
*0643 23            013D >          INX   H          ; RAM
*0644 CD 96 04      013E >          CALL  BCVDE      ; ENDE ?
*0647 DA 1E 06      013F >          JC    HILF11     ; NEIN
*064A 3E 9B         0140 >          MVI   A,9BH      ; ALLES EINGANG UND NULL
*064C D3 03         0141 >          OUT   SBUS       ; ALLES TRISTATE
*064E C3 61 04      0142 >          JMP   ENDE       ; FERTIG
*0651 FE 07         0143 >HILF12    CPI   07H        ; CODE 07 ?
*0653 C2 7B 06      0144 >          JNZ   HILF14     ; NEIN
```

```
                        0145 >; CODE=07: EPROM-RAM VERGLEICHEN BC=ANF DE=END HL=RAM
*0656 13                0146 >          INX   D          ; ENDADRESSE + 1
*0657 CD 96 04          0147 >          CALL  BCVDE      ; VERGLEICHEN
*065A D2 52 04          0148 >          JNC   ANKEN      ; FEHLER: BC GROESSER DE
*065D 3E 82             0149 >          MVI   A,82H      ; A=AUS B=EIN(DATEN) C=AUS
*065F D3 03             014A >          OUT   SBUS       ; DATENEINGABE
*0661 3E 98             014B >          MVI   A,98H      ; 100 11 000
*0663 D3 02             014C >          OUT   ABUSH      ; VPP=VCC=5V  OE=CE=HIGH
*0665 CD A9 04          014D >HILF13    CALL  DEIN       ; DATENLESEN
*0668 BE                014E >          CMP   M          ; MIT SPEICHER VERGLEICHEN
*0669 C4 70 04          014F >          CNZ   FEHLG      ; UNGLEICH: FEHLER
*066C 03                0150 >          INX   B          ; EPROM-ADRESSE + 1
*066D 23                0151 >          INX   H          ; RAM-ADRESSE + 1
*066E CD 96 04          0152 >          CALL  BCVDE      ; ENDE ?
*0671 DA 65 06          0153 >          JC    HILF13     ; NEIN
*0674 3E 9B             0154 >          MVI   A,9BH      ; ALLES EINGANG
*0676 D3 03             0155 >          OUT   SBUS       ; AUSGÄNGE TRISTATE
*0678 C3 61 04          0156 >          JMP   ENDE       ; FERTIG
*067B FE 08             0157 >HILF14    CPI   08H        ; CODE 08 ?
*067D C2 A6 06          0158 >          JNZ   HILF16     ; NEIN
                        0159 >; CODE=8: EPROM NACH RAM BC=ANF DE=END HL=RAM
*0680 13                015A >          INX   D          ; ANFANGSADRESSE + 1
*0681 CD 96 04          015B >          CALL  BCVDE      ; VERGLEICHEN
*0684 D2 52 04          015C >          JNC   ANKEN      ; FEHLER BC GROESSER DE
*0687 3E 82             015D >          MVI   A,82H      ; A=AUS B=EIN(DATEN) C=AUS
*0689 D3 03             015E >          OUT   SBUS       ; DATENEINGABE
*068B 3E 98             015F >          MVI   A,98H      ; 100 11 000
*068D D3 02             0160 >          OUT   ABUSH      ; VPP=VCC=5V  OE=CE=HIGH
*068F CD A9 04          0161 >HILF15    CALL  DEIN       ; BYTE LESEN
*0692 77                0162 >          MOV   M,A        ; SPEICHERN
*0693 BE                0163 >          CMP   M          ; VERGLEICHEN
*0694 C4 70 04          0164 >          CNZ   FEHLG      ; UNGLEICH: FEHLER
*0697 03                0165 >          INX   B          ; EPROM-ADRESSE + 1
*0698 23                0166 >          INX   H          ; RAM-ADRESSE + 1
*0699 CD 96 04          0167 >          CALL  BCVDE      ; ENDE ?
*069C DA 8F 06          0168 >          JC    HILF15     ; NEIN
*069F 3E 9B             0169 >          MVI   A,9BH      ; ALLES EINGANG
*06A1 D3 03             016A >          OUT   SBUS       ; AUSGÄNGE TRISTATE
*06A3 C3 61 04          016B >          JMP   ENDE       ; FERTIG
*06A6 C3 43 04          016C >HILF16    JMP   CODE       ; UNZULAESSIGER CODE
E0000                   016D >          END
```

Aufgabe:

Datum:

Seite:

Zeile	Adresse	Inhalt				Name	Befehl	Operand	Bemerkung
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									

Aufgabe:

Datum:

Seite:

Zeile	Adresse	Inhalt				Name	Befehl	Operand	Bemerkung
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									

VI G

MVUS-Platine: RES-Taste: Programmabbruch mit Meldung MINITOR 2.0
STP-Taste: Programmabbruch mit Ausgabe der Register
TAS-Taste: Funktion über Brückenfeld einstellbar

Monitorkommandos:

:S A>adresse (cr)	Speicherbytes anzeigen und ändern
	Unterkommandos
	aa- nn neue Eingabe
	aa- (lz) Adresse + 1
	aa- - Adresse - 1
	aa- (cr) Kommandoende
:D A>anfang ␣ E>ende (cr)	Speicherbereich anzeigen
:F A>anfang ␣ E>ende ␣ B>byte (cr)	Speicherbereich mit Byte füllen
:G S>adresse (cr)	Benutzerprogramm starten ohne Haltepunkt
:G S>adresse (lz) H>adresse (cr)	Benutzerprogramm starten mit Haltepunkt
:T S>adresse (cr)	Benutzerprogramm im Einzelschritt verfolgen

*STOP: Programm mit STP angehalten, weiter mit :G oder :T
*HPKT: Programm hat Haltepunkt erreicht, weiter mit :G oder :T
*EBEF: Programmverfolgung im Einzelschritt, weiter :lz oder :G
Programmabbruch mit :(cr)

:E (cr) Editor aufrufen

EDITOR 1.0 Alt/Neu? A/N> A oder N eingeben

A/N> N neues Programm eingeben
altes Programm verloren

EINGABE
#001> es wirken die Tasten
← Cursor links
→ Cursor rechts
Tab Tabulator

zeile (cr) Zeile abschicken

#xxx> (cr) in Spalte 1:
Ende des Eingabebetriebs
nach Kommandobetrieb

A/N> A altes Programm ändern
Programm muß vorhanden sein

KOMMANDO
#001> 1. alte Zeile
KOMM> Kommandos:

#nr (cr)	Zeile nr positionieren
B (cr)	1. Zeile positionieren
I (cr)	alle Zeilen ausgeben
In (cr)	n Zeilen ausgeben
I (cr)	hinter pos. Zeile einfügen
	Eingabebetrieb; neue Zeilen
D (cr)	pos. Zeile löschen
(cr)	nächste Zeile positionieren
- (cr)	vorhergehende Zeile pos.
E (cr)	Ende des Editors
A (cr)	Assembler aufrufen

:A (cr) Assembler aufrufen

ASSEMBLER 1.0 LADEN? J/N> LISTE? J/N> ZEILEN/SEITE > HALT? J/N>

Bildschirm ausgeben: Print-Taste
Drucker einschalten: Setup-Taste oder Setup Ctrl Shift
mit Cursor nach AUX OFF
mit Leertaste AUX ON dann wieder Setup

PORT 10H

PORT 18H

Adressen

Daten

U19 8255 Parallele E/A

A-Port: 00H
B-Port: 01H
C-Port: 02H
S-Reg: 03H

(2716 EPROM)
U17 Programmiersockel

Speicher

U20 2716/6116/2016

U18 2716/6116/2016

U14 2716 MONITOR

U13

U1 8085 Mikroprozessor

U23 8155 RAM/IO/TIMER

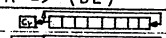
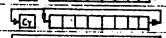
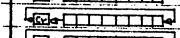
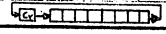
U16 74 154

Serielle E/A

MVUS 8085
Übersichtsschaltplan

FH Dib VI G 7/83 S

C 225

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
LDA	adresse	A <= Speicher	3A	3	13	
STA	adresse	A => Speicher	32	3	13	
IN	port	A <= Periph.	DB	2	10	
OUT	port	A => Periph.	D3	2	10	
ORA	A	teste Akku	B7	1	4	xx 0 x 0
XRA	A	lösche Akku	AF	1	4	0 1 0 0 0
LDAX	B	A <= (BC)	0A	1	7	
LDAX	D	A <= (DE)	1A	1	7	
STAX	B	A => (BC)	02	1	7	
STAX	D	A => (DE)	12	1	7	
RLC			07	1	4	x
RRC			0F	1	4	x
RAL			17	1	4	x
RAR			1F	1	4	x

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
MVI	A, kon	A <= konstante	3E	2	7	
CPI	konst	A - konstante	FE	2	7	xx x x x
ADI	konst	A <= A + konst	C6	2	7	xx x x x
ACI	konst	A <= A + kon+Cy	CE	2	7	xx x x x
SUI	konst	A <= A - konst	D6	2	7	xx x x x
SBI	konst	A <= A - kon-Cy	DE	2	7	xx x x x
ANI	konst	A <= A UND kon	E6	2	7	xx 1 x 0
ORI	konst	A <= A ODR kon	F6	2	7	xx 0 x 0
XRI	konst	A <= A XOR kon	EE	2	7	xx 0 x 0
CMA		A <= NICHT A	2F	1	4	
DAA		A <= BCD-Korr.	27	1	4	xx x x x
STC		Carry <= 1	37	1	4	1
CMC		Carry <= Carry	3F	1	4	x
ORA	A	Carry <= 0	B7	1	4	xx 0 x 0

Bef.	Operand	Wirkung	A OP	A B	A T	B OP	B B	B T	C OP	C B	C T	D OP	D B	D T	E OP	E B	E T	H OP	H B	H T	L OP	L B	L T	M OP	M B	M T	Bedingung SZxHOPvCy
MVI	reg,kon	reg <= konst	3E	2	7	06	2	7	0E	2	7	16	2	7	1E	2	7	26	2	7	2E	2	7	36	2	10	
MOV	A, reg	A <= register	7F	1	4	78	1	4	79	1	4	7A	1	4	7B	1	4	7C	1	4	7D	1	4	7E	1	7	
MOV	B, reg	B <= register	47	1	4	40	1	4	41	1	4	42	1	4	43	1	4	44	1	4	45	1	4	46	1	7	
MOV	C, reg	C <= register	4F	1	4	48	1	4	49	1	4	4A	1	4	4B	1	4	4C	1	4	4D	1	4	4E	1	7	
MOV	D, reg	D <= register	57	1	4	50	1	4	51	1	4	52	1	4	53	1	4	54	1	4	55	1	4	56	1	7	
MOV	E, reg	E <= register	5F	1	4	58	1	4	59	1	4	5A	1	4	5B	1	4	5C	1	4	5D	1	4	5E	1	7	
MOV	H, reg	H <= register	67	1	4	60	1	4	61	1	4	62	1	4	63	1	4	64	1	4	65	1	4	66	1	7	
MOV	L, reg	L <= register	6F	1	4	68	1	4	69	1	4	6A	1	4	6B	1	4	6C	1	4	6D	1	4	6E	1	7	
MOV	M, reg	M <= register	77	1	7	70	1	7	71	1	7	72	1	7	73	1	7	74	1	7	75	1	7				
INR	reg	reg <= reg + 1	3C	1	4	04	1	4	0C	1	4	14	1	4	1C	1	4	24	1	4	2C	1	4	34	1	10	xx x x
DCR	reg	reg <= reg - 1	3D	1	4	05	1	4	0D	1	4	15	1	4	1D	1	4	25	1	4	2D	1	4	35	1	10	xx x x
CMP	reg	A - register	BF	1	4	B8	1	4	B9	1	4	BA	1	4	BB	1	4	BC	1	4	BD	1	4	BE	1	7	xx x x x
ADD	reg	A<= A + reg	87	1	4	80	1	4	81	1	4	82	1	4	83	1	4	84	1	4	85	1	4	86	1	7	xx x x x
ADC	reg	A<= A + reg+Cy	8F	1	4	88	1	4	89	1	4	8A	1	4	8B	1	4	8C	1	4	8D	1	4	8E	1	7	xx x x x
SUB	reg	A<= A - reg	97	1	4	90	1	4	91	1	4	92	1	4	93	1	4	94	1	4	95	1	4	96	1	7	xx x x x
SBB	reg	A<= A - reg-Cy	9F	1	4	98	1	4	99	1	4	9A	1	4	9B	1	4	9C	1	4	9D	1	4	9E	1	7	xx x x x
ANA	reg	A<= A UND reg	A7	1	4	A0	1	4	A1	1	4	A2	1	4	A3	1	4	A4	1	4	A5	1	4	A6	1	7	xx 1 x 0
ORA	reg	A<= A ODER reg	B7	1	4	B0	1	4	B1	1	4	B2	1	4	B3	1	4	B4	1	4	B5	1	4	B6	1	7	xx 0 x 0
XRA	reg	A<= A XOR reg	AF	1	4	A8	1	4	A9	1	4	AA	1	4	AB	1	4	AC	1	4	AD	1	4	AE	1	7	xx 0 x 0

Bef.	Operand	Wirkung	B OP	B B	B T	D OP	D B	D T	H OP	H B	H T	SP OP	SP B	SP T	PSW(A,be) OP	PSW(A,be) B	PSW(A,be) T	Bedingung SZxHOPvCy
LXI	reg,kon	rp <= 16-Bit	01	3	10	11	3	10	21	3	10	31	3	10				
INX	register	rp <= rp + 1	03	1	6	13	1	6	23	1	6	33	1	6				
DCX	register	rp <= rp - 1	0B	1	6	1B	1	6	2B	1	6	3B	1	6				
DAD	register	HL <= HL + rp	09	1	10	19	1	10	29	1	10	39	1	10				x
PUSH	register	rp => Stapel	C5	1	12	D5	1	12	E5	1	12				F5	1	12	
POP	register	rp <= Stapel	C1	1	10	D1	1	10	E1	1	10				F1	1	10	

Interruptregister RIM:							
7	6	5	4	3	2	1	0
SID	7.5	6.5	5.5	INT	7.5	6.5	5.5
Anzeigen				FF	Masken		

Interruptregister SIM:							
7	6	5	4	3	2	1	0
SOD	1	x	7.5	1	7.5	6.5	5.5
0				FF	0	Masken	

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
LHLD	adresse	L<=adr H<=adr+1	2A	3	16	
SHLD	adresse	L=>adr H=>adr+1	22	3	16	
XCHG		HL <=> DE	EB	1	4	
XTHL		HL <=> Stapel	E3	1	16	
PCHL		PC <= HL	E9	1	6	
SPHL		SP <= HL	F9	1	6	

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
NOP		keine	00	1	4	
RIM		A <= Int.Reg.	20	1	4	
SIM		A => Int.Reg.	30	1	4	
EI		Interrupt frei	FB	1	4	
DI		Interrupt gesp	F3	1	4	
HLT		Proz. anhalten	76	1	5	

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
JMP	adresse	springe immer	C3	3	10	
JZ	adresse	sprg. bei = 0	CA	3	7/10	
JNZ	adresse	sprg. bei ≠ 0	C2	3	7/10	
JC	adresse	sprg. bei Cy=1	DA	3	7/10	
JNC	adresse	sprg. bei Cy=0	D2	3	7/10	
JM	adresse	sprg. bei S =1	FA	3	7/10	
JP	adresse	sprg. bei S =0	F2	3	7/10	
JPE	adresse	sprg. bei P =1	EA	3	7/10	
JPO	adresse	sprg. bei P =0	E2	3	7/10	

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
RST	0	Start bei 0000	C7	1	12	
RST	1	Start bei 0008	CF	1	12	
RST	2	Start bei 0010	D7	1	12	
RST	3	Start bei 0018	DF	1	12	
RST	4	Start bei 0020	E7	1	12	
RST	5	Start bei 0028	EF	1	12	
RST	6	Start bei 0030	F7	1	12	
RST	7	Start bei 0038	FF	1	12	
Alle RST-Bef.:		PC => Stapel				

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
CALL	adresse	rufe Unterprg.	CD	3	18	
CZ	adresse	rufe bei = 0	CC	3	9/18	
CNZ	adresse	rufe bei ≠ 0	C4	3	9/18	
CC	adresse	rufe bei Cy= 1	DC	3	9/18	
CNC	adresse	rufe bei Cy= 0	D4	3	9/18	
CM	adresse	rufe bei S = 1	FC	3	9/18	
CP	adresse	rufe bei S = 0	F4	3	9/18	
CPE	adresse	rufe bei P = 1	EC	3	9/18	
CPO	adresse	rufe bei P = 0	E4	3	9/18	

Bef.	Operand	Wirkung	OP	B	T	Bedingung SZxHOPvCy
RET		immer zurück	C9	1	10	
RZ		zur. bei = 0	C8	1	6/12	
RNZ		zur. bei ≠ 0	C0	1	6/12	
RC		zur. bei Cy= 1	D8	1	6/12	
RNC		zur. bei Cy= 0	D0	1	6/12	
RM		zur. bei S = 1	F8	1	6/12	
RP		zur. bei S = 0	F0	1	6/12	
RPE		zur. bei P = 1	E8	1	6/12	
RPO		zur. bei P = 0	E0	1	6/12	

Beschreibung zu MVUS.C

Einleitung

Das Programm MVUS.C bietet die Möglichkeit das Mikrocomputer Versuchs- und Übungssystem MVUS 8085 an einen IBM AT anzuschließen. Das MVUS wird dazu über die serielle Schnittstelle mit dem IBM verbunden.

Das Programm kann in 4 Modi betrieben werden:

Modus 0	In diesem Modus arbeitet der IBM als reines Terminal mit dem MVUS. Die Möglichkeiten entsprechen denen des Qume Terminals.
Modus 1	Betrieb mit der Editor-Assemblerkarte des MVUS. Bei eingesteckter Assemblerkarte können Programme auf dem IBM erstellt und als Textfiles zum MVUS übertragen werden. Dort stehen Sie dem Assembler und dem Editor des MVUS zur Verfügung. Zusätzlich können Programme vom MVUS zum IBM übertragen und dort als Textfiles abgespeichert werden.
Modus 2	Betrieb mit dem CPM-Emulator KCE. Programme werden vollständig mit dem IBM entwickelt. Daher kann die Assemblerkarte des MVUS nun entfallen. Es werden nur noch die entstandenen Binärfiles übertragen, welche dann auf dem MVUS ablauffähig sind.
Modus 3	Übertragen von Binärfiles zum MVUS. Dieser Modus ist für die Anwendung in Batch-Dateien vorgesehen.

Alle Funktionen, die über den reinen Terminalbetrieb hinausgehen, werden mit Hilfe der Funktionstasten aufgerufen.

Voraussetzungen

Hardware

Voraussetzung ist ein IBM AT/XT oder ein kompatibler Rechner mit angeschloßenem Epson FX80 kompatiblen Drucker und einer freien seriellen Schnittstelle.

Software

Vor dem Start von MVUS sollten sich im Unterverzeichnis CPM auf der Festplatte folgende Programme befinden:

MVUS.EXE	(Dieses Programm)
KCE.EXE	(CPM-Emulator)
M80.COM	(8085-Assembler)
L80.COM	(8085-Linker)
DDT.COM	(8085-Debugger)
SUBMIT.COM	(Batch-Verarbeitung unter CPM)
AS8085.SUB	(CPM Batchfile zum Aufruf des Assemblers und Linkers)
AS85.BAT	(Batchfile zum kompletten Übersetzen eines Programms)
SKELMAC.MAC	(Skelett zur Erstellung eines Programms)
MAC.PRF	(Profile für den IBM-Professional- Editor)

Der IBM-Professional-Editor "EDIT" muß sich im Suchpfad (DOS-Befehl PATH) befinden. Das Programm KCE muß vollständig installiert sein, d.h. in der config.sys müssen die Zeilen

```
device = ndrive.sys -ra -u0
device = cdrive.sys -ra -u0
device = ansi.sys
```

und in der autoexec.bat die Zeilen

```
set kcemode=soft   (bzw.=z8085 bei vorhandener Z8085-Karte)
set kcevideo=ibm
set kcedrives=F:-A:
waehle -lF pc
```

aufgenommen werden.

Starten von MVUS

Zum Starten von MVUS wechselt man in das Verzeichnis CPM und ruft dann MVUS folgendermaßen auf:

MVUS [/mx] [/cy] [/lz] [Filename.Ext]

Dabei haben die Parameter folgende Bedeutung:

/mx Bestimmt den Modus, in welchem das Programm betrieben wird. X muß einen Wert zwischen 0 und 3 haben. Defaultwert ist 2.

/cy Bestimmt die zu verwendende serielle Schnittstelle. Y muß einen Wert zwischen 1 und 4 haben. Defaultwert ist 1, damit wird COM1: zur Verbindung mit dem MVUS benutzt.

/lz Bestimmt den zu verwendenden Druckerport. Z muß den Wert 1 oder 2 haben. Defaultwert ist 1, damit wird LPT1: zur Druckausgabe benutzt.

In eckige Klammern gesetzte Parameter sind optional. Wird Modus 3 gewählt, so muß der Name der auszugebenden Datei mit angegeben werden (mit Erweiterung). Ist der MVUS ordnungsgemäß mit dem IBM verbunden, so erscheint eine Überschrift, eine Information in welchem Modus man sich befindet und das Bereitschaftszeichen des Minitors (MINITOR 2.0).

Betrieb von MVUS

Allgemeines

Wie schon oben erwähnt werden die verschiedenen Funktionen des Programms durch Funktionstasten aufgerufen. Die **Funktiontasten** sollten nur dann verwendet werden, wenn das MVUS seinen Bereitschaftstext (MINITOR 2.0) zeigt, d.h. **keine Verwendung aus dem Editor des MVUS heraus.**

Das Programm arbeitet mit einem **Workfile**, d.h. beim Aufruf verschiedener Funktionen ist es nicht nötig immer wieder den Namen des zu bearbeitenden Programmes anzugeben. Stattdessen wird jeweils der Workfilename verwendet. Wurde noch kein Workfilename angegeben, so wird man beim Aufruf einer Funktion zunächst nach einem Namen gefragt. Der Name muß der DOS-Spezifikation genügen. Wird der Name ohne Erweiterung angegeben, so wird für Programmtexte automatisch .MAC verwendet. Der angegebene Name sollte keinen Pfad enthalten, da der 8085-Assembler und Linker keine Pfade verarbeiten können. Gibt man keinen Namen an, sondern drückt nur RETURN, so kommt man zum MINITOR zurück.

Ausgaben auf den **Drucker** erreicht man durch drücken der Tastenkombination **Control-P**. Alles, was auf den Bildschirm ausgegeben wird, erscheint nun auch auf dem Drucker. Durch nochmaliges drücken von Control-P wird die Ausgabe auf den Drucker wieder beendet. Zum Drucken von Textfiles steht in den Modi 1 und 2 zusätzlich eine Funktionstaste zur Verfügung (siehe unten).

Drei Funktionstasten haben in jedem Modus dieselbe Funktion:

F1 bringt einen kurzen Hilfetext auf den Bildschirm, der die Belegung der Funktionstasten im aktuellen Modus zeigt.

Alt-F1 schaltet einen Modus weiter, wobei Modus 3 (nur Senden einer Datei) nicht angewählt werden kann. Von Modus 2 wird zu Modus 0 gewechselt.

F10 beendet das Programm und kehrt zu DOS zurück.

Betrieb im Modus 0

Dieser Modus simuliert das Qume Terminal. Alle Befehle arbeiten so, wie es in den Unterlagen für den Terminal-Betrieb des MVUS beschrieben wird.

Betrieb im Modus 1

Dieser Modus setzt voraus, daß die Editor-Assemblerkarte im MVUS steckt. Zusätzlich zu den Möglichkeiten, die der normale Terminalbetrieb gestattet sind folgende Funktionen verfügbar:

F1 Bringt die Belegung der Funktionstasten auf den Bildschirm.

F2 Druckt Textdateien aus. Bei der Frage nach dem Dateinamen wird der Name der zu druckenden Datei *mit Erweiterung* angegeben. Die folgende Frage nach der Anzahl der Zeichen pro Zeile beim Druck ist hauptsächlich für List-Files des Assemblers gedacht, die meist mit 132 Zeichen/Zeile gedruckt werden. Der Druck kann jederzeit durch drücken einer beliebigen Taste abgebrochen werden.

F3 Diese Taste ruft den IBM-Professional-Editor "EDIT" auf. Als Parameter wird EDIT der Workfilename übergeben. Nach dem Editieren des Workfiles wird es automatisch zum MVUS übertragen, wo es dann dem Assembler und Editor des MVUS zur Verfügung steht. EDIT arbeitet mit der Datei MAC.PRF in der Tabpositionen, Großschrift und maximale Zeilenlänge definiert sind. Der Editor des MVUS kann nur Zeilen bis zu einer Länge von 60 Zeichen bearbeiten. Längere Zeilen werden bei der Übertragung zum MVUS abgeschnitten, Tabulatoren werden dem MVUS-Editor angepaßt.

F4 Ein neuer Workfilename wird eingelesen. Alle folgenden Aufrufe durch die Funktionstasten werden mit diesem Namen durchgeführt.

F5 Ruft wie F3 EDIT auf, ohne jedoch die Datei nach Verlassen des Editors gleich zum MVUS zu senden. Auf diese Weise kann man auch Dateien editieren, die keine Assemblerprogramme für das MVUS sind.

F6 Die Textdatei, deren Name als Workfile angegeben wurde, wird zum MVUS übertragen, um dort weiterverarbeitet zu werden.

F7 Das Programm welches gerade im Editor des MVUS steht, wird in eine Datei übertragen deren Name mit Erweiterung angegeben werden muß. *Diese Funktion sollte wirklich nur dann aufgerufen werden, wenn auch wirklich ein Programm im Editor des MVUS steht, da es sonst zum Systemabsturz kommen kann!*

F8 Nicht belegt.

F9 Nicht belegt.

Betrieb im Modus 2

Dieser Modus ermöglicht es 8085-Programme ganz auf dem IBM zu entwickeln. Programme ohne IN- und OUT-Befehle können mit Hilfe des Debuggers DDT vollständig getestet werden. Die entstandenen Programme können zum MVUS übertragen werden und sind dort ablauffähig. Um die Programme zum MVUS übertragen zu können müssen folgende Regeln für Assemblerprogramme eingehalten werden:

1.) Vor jedem ORG-Befehl muß die Assemblerdirektive ASEG stehen, damit der Linker das Programm an absolute Adressen lädt.

2.) Die erste ORG-Anweisung, die den Beginn des Codes festlegt muß größer oder gleich **103H** sein. (Beim MVUS meist 1000H).

3.) Die END-Anweisung muß als Argument ein Label aufweisen, das auf den ersten Befehl nach der in 2.) festgelegten Startadresse zeigt. (z.B. END START)

4.) Alle Daten und jeglicher Code müssen sich an höheren Adressen als der durch die in 2.) festgelegten Startadresse befinden.

Weiter ist zu beachten, daß der M80-Assembler keine Umlaute verarbeitet (auch nicht im Kommentar!) und grundsätzlich alle Eingaben in Großschrift erwartet.
Listfiles des M80-Assembler sollten die Erweiterung .LST besitzen und nicht die Standarderweiterung .PRN, da es leicht zu Konflikten mit dem IBM- Professional-Editor kommen kann, der die Erweiterung .PRN für Druckausgaben benutzt.

Belegung der Funktionstasten im Modus 2:

F1 Bringt die Belegung der Funktionstasten auf den Bildschirm.

F2 Druckt Textdateien (wie in Modus 1).

F3 Ruft den Editor "EDIT" mit dem Workfile als Parameter auf.

F4 Liest einen neuen Workfilenamen ein (wie im Modus 1).

F5 Ruft den M80-Assembler **ohne** Workfile als Parameter auf. Der Assembler verlangt Großschreibung. Ein Programm sollte mit der Befehlsfolge "*,NAME.LST=NAME " assembliert werden, um ein Listfile mit der Erweiterung .LST zu erhalten. Soll das Programm zum MVUS übertragen werden, so muß auf die oben genannten Punkte geachtet werden.

F6 Ruft den Linker L80 **ohne** Workfile als Parameter auf.Auch hier gilt: der Linker erwartet Eingaben in Großschrift.

F7 Sendet den durch assemblierten und gelinkten Code zum MVUS. Es wird die Datei mit dem Workfilenamen und der Erweiterung .COM übertragen. Wird keine Datei mit diesem Namen gefunden oder genügt sie nicht den oben genannten Anforderungen, so wird eine Fehlermeldung ausgegeben.

F8 Ruft nacheinander den Assembler und Linker auf und überträgt das entstandene .COM-File zum MVUS. Es werden alle erforderlichen Dateien des Workfiles erstellt, einschließlich des Listfiles mit der Erweiterung .LST. Fehler bei der Assemblierung werden leider nicht entdeckt, so daß man bei der Übersetzung selbst auf Fehlermeldungen achten muß.

F9 Ruft den Debugger DDT mit dem Workfilenamen und der Erweiterung .COM auf. Damit kann das Programm noch auf der Ebene des IBM getestet werden, indem man es z.B. im Einzelschritt durchläuft. OUT- und IN-Befehle für die Anzeigen und Leuchtdioden des MVUS werden dabei natürlich nicht entsprechend verarbeitet. Dafür steht der Einzelschrittbetrieb des MVUS-MINITORS zur Verfügung.

Betrieb mit Modus 3

Von DOS aus wird MVUS mit dem Parameter /m3 und dem Namen des zu übertragenden COM-Files aufgerufen. Entspricht das Quell-File den oben genannten 4 Regeln, so wird das COM-File zum MVUS gesandt, wo es gestartet werden kann.

Erstellen von Programmen

Als Beispiel soll ein Programm mit dem Namen MUL.MAC erstellt werden.

Im Batch-Betrieb auf DOS-Ebene

```
C:\>cd cpm                ;Ins Unterverzeichnis CPM wechseln
C:\CPM>edit skelmac.mac   ;EDIT aufrufen, als OUTFILE MUL.MAC angeben
                          ;Programmtext eingeben, zu DOS zurückkehren
C:\CPM>as85 mul           ;Batchfile mit Programmnamen ohne Erweiterung aufrufen
```

Wenn keine großen Fehler aufgetreten sind, bekommt man die MINITOR- Meldung. Das Programm kann getestet werden. Es sind die Dateien MUL.MAC, MUL.REL, MUL.LST und MUL.COM entstanden. Beenden des MVUS mit F10.

Im Modus 2 vom Programm MVUS

```
C:\>cd cpm                ;Ins Unterverzeichnis CPM wechseln
C:\CPM>mvus               ;MVUS im Default-Modus 2 aufrufen
                          ;F3 drücken, mul als Workfilenamen angeben
                          ;Beim Editor INPUTFILE in SKELMAC.MAC umbenennen
                          ;Beim Editor OUTPUTFILE MUL.MAC angeben
                          ;Programm erstellen und zum MINITOR zuückkehren
                          ;F8 drücken, MUL wird übersetzt und übertragen
```

Das Programm kann getestet werden. Es sind die Dateien MUL.MAC, MUL.REL, MUL.LST und MUL.COM entstanden.

Inhalt der Batchdateien

Inhalt von AS85.BAT:

```
kce submit as8085 %1
if errorlevel 1 goto :quit
mvus /m3 %1.com
if errorlevel 1 goto :quit
mvus /m2
:quit
```

Inhalt von AS8085.SUB:

```
M80    $1,$1.LST=$1/L/R
L80    $1,$1/N/E
```

Inhalt von SKELMAC.MAC:

```
;SKELETT ZUR ERSTELLUNG VON PROGRAMMEN FUER DEN MVUS-8085.
        ASEG
        ORG   1000H    ;STARTADRESSE FUER MVUS-PROGRAMME
;EINFUEGEN DER BEFEHLSFOLGE AN DIESER STELLE:
;
START:

ENDE:   RST   00H      ;ZURUECK ZUM BETRIEBSSYSTEM
;EINFUEGEN DER DATEN AN DIESER STELLE
;ORG KANN NEU GESETZT WERDEN, MUSS ABER > ALS DIE OBEN
;FESTGELEGTE ADRESSE SEIN!

        END   START    ;ENDE DES ASSEMBLERPROGRAMMS
```

Konfiguration von EDIT abgelegt in MAC.PRF:

The IBM Personal Computer - Professional Editor

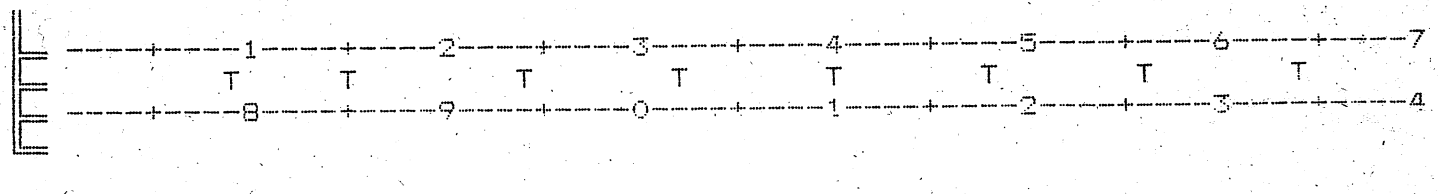
Profile Options

READ ONLY	NO	UPPER CASE	YES	VT SCROLL INC	12
EDIT IN PLACE	NO	EXTEND ASCII	NO	HZ SCROLL INC	20
BACKUP FILE	NO	AUTO INDENT	NO	RESERVE [Kb]	4
READ INPUT	YES	MASK INPUT	NO	LEFT BOUND	1
RE-CYCLE	NO	AUTO SCROLL	YES	RIGHT BOUND	140
LEFT MARGIN	0	WORD WRAPPING	YES	PRINTER UNIT	0
RIGHT MARGIN	0	AUDIO LEVEL	2	CHARS/LINE	80
PARA INDENT	0	WRITE VERIFY	0	LINES/INCH	6

The IBM Personal Computer - Professional Editor

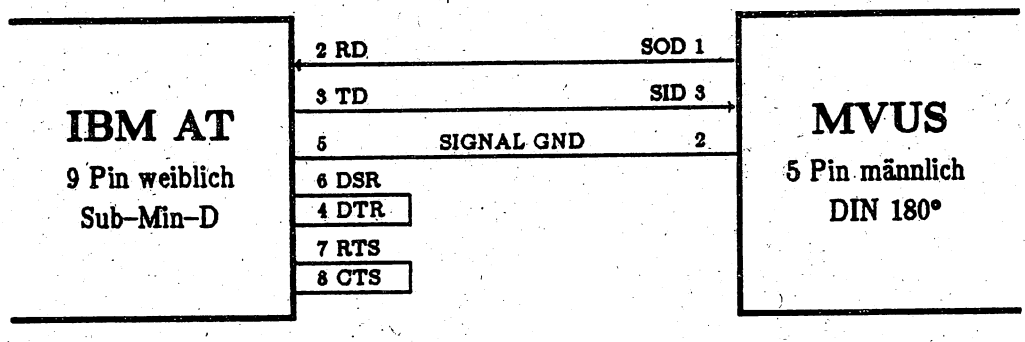
Tabs & Mask Options

TAB STOPS: (BLANK for no stop, NON-BLANK for stop)



Verbindung IBM AT — MVUS 8085

Die Verbindung zwischen den beiden Rechnern wird über die seriellen Schnittstellen mit Hilfe eines dreiadrigen Kabels geschaffen. Für die Verbindung benötigt man IBM–seitig eine 9-polige Subminiatur-D-Buchse und MVUS–seitig einen 5-poligen 180° DIN-Stecker. Da kein Hardware–Handshake verwendet wird, muß auf der Seite des IBM durch Brücken dafür gesorgt werden, daß die Handshake–Leitungen die für den Betrieb nötigen Pegel bekommen. Die Verdrahtung erfolgt nach folgender Skizze:



Die Zahlen sind die Pin–Nummern der Stecker. Es bedeuten:

RD	Receive Data
TD	Transmitt Data
DSR	Data Set Ready
DTR	Data Terminal Ready
RTS	Request To Send
CTS	Clear To Send
SOD	Serial Output Data
SID	Serial Input Data
GND	Masse

```
/* MVUS.C                                                              */
/* Interruots nur auf der Empfangsseite - die Sendeseite wird gepollt. */
/* Abbruch durch Eingabe von F10.                                      */
/* Nur Dreidrahtverbindung (keine Modemsignale)!                       */
/* Version fuer TurboC V1.0  - benoetigt kein Assembler Programm       */
/* Erstellt fuer die Bedienung des MVUS Michael Meyer 1/88  FH Dieburg */

/*          COMPILIEREN MIT SPEICHERMODEL COMPACT !!                   */

#include <stdio.h>
#include <process.h>
#include <ctype.h>
#include <sys\stat.h>
#include <getopt.c>     /* C-Programm, wird mit TurboC geliefert       */


#define EDIT   "edit"    /* Name des benutzten Editors                 */
#define LST    ".LST="   /* Erweiterungsname von m80-Listfiles          */
#define LBUF   50000     /* Groesse des Einlese-Ringpuffers
                            = 800 Zeilen Programm                      */
#define VERZ   11000     /* Verzoegerung bei Befehlsausgabe            */
                         /* 5000 bei XT, 11000 bei AT                  */
#define PLEN   250       /* Laenge des Dateilesepuffers fuer eine Zeile*/
#define TABANZ 0x08      /* Anzahl der Spaces bei Tab                  */
#define TAB    0x09      /* Tabulatorzeichen (ASCII-Code)              */
#define LF     0x0a
#define CR     0x0d
#define ESC    0x1b
#define FALSE  0
#define DRK80  "\x12\x1bP" /* Cancels Condensed,Select Pica            */
#define DRK132 "\x0f"      /* Select Condensed                         */

#define ANFTEX 0x4400    /* Anfang des Textbereiches auf Assemblerkarte MVUS */
#define ANFDAT 0x403c    /* Anfang des Datenbereichs des MVUS Editors  */
#define ZLEN   0x3e      /* Zeilenlaenge des MVUS-Editors              */
#define MAXLEN 60        /* Maximale Text-Zeilenlaenge des MVUS-Editors*/

/* Uebertragungs-Parameter :                                           */
#define BAUDLSB 0x18
#define BAUDMSB 0x00     /* -> 4800 BAUD                               */
#define UEPAR   0x03     /* 8 Daten, 2 Stop, no parity                 */

#define CHAR unsigned char

#define EOI    0x64      /* end of spec. interr. 4                     */
#define OCW2   0x20      /* oper. control word on 8259                 */
#define IER    0x9       /* interr. enable reg.                        */
#define IIR    0xa       /* interr. ident. reg.                        */
#define MCR    0xc       /* modem control reg.                         */
#define RBR    0x8       /* receive buffer reg.                        */
#define LSR    0xd       /* line status reg.                           */
#define MSR    0xe       /* modem status reg.                          */
#define LCR    0xb       /* line control reg.                          */
#define SERINT 0x0c      /* int number                                 */
#define CLRSCR "\033[2J" /* ANSI-Sequenz, erfordert ANSI.SYS !         */
#define AF1    104-59    /* Alt-F1-Code                                */

typedef enum {f1,f2,f3,f4,f5,f6,f7,f8,f9,f10} funct;
```

```
/* Globale Variablen :                                                    */

CHAR            c,recbuf[LBUF],*radr,*wadr,workfile[30],outfile[30],wfohne[30],
                empfang,ckb,puffer[PLEN],bef[30],*pui,*anfpui,*ifile,lst[5];
char            com,modus;
unsigned int    adresse,zeile,ende,k,i,aktpos,laenge;
int             o8259,comx,cb,opt,ret;
void interrupt  (*oldvect)();
FILE            *fl;


/***************************** SERVROUT ***************************************/
/* Interruptservice Routine                                                   */
/* Empfaengt Zeichen und reiht es in den Ringpuffer an wadr ein               */

void interrupt servrout()
{
CHAR ic;
  if (inportb(IIR+comx)!=0x04) goto servr8; /* Woher stammt der Interrupt?   */

  if ((inportb(LSR+comx)&0x01)==0) goto servr8; /* Receive data bit gesetzt? */
  ic = inportb(RBR+comx);                 /* Zeichen lesen                  */
  if(empfang=='f') goto servr8;           /* Empfang abgeschaltet ?         */
  if (wadr<=&recbuf[LBUF-1]) *wadr++=ic;  /* Ringpuffer voll ->nein, ablegen */
  else
  {
    wadr = recbuf;                        /* ja-> Zeiger auf Anfang         */
    *wadr++ = ic;                         /* ablegen                        */
   }
  servr8:
  outportb(OCW2,EOI);                     /* 8259 Ende des Interrupts mitteilen */
}                                         /* end of servrout                */


/***************************** INSTALL ****************************************/
/* Interruptvektor anlegen und 8250 programmieren                             */

void install()
{
  outportb(MCR+comx,0);                   /* Reset fuer UART 8250           */
  inportb(LSR+comx);
  inportb(RBR+comx);
  inportb(MSR+comx);
  disable();                              /* Interupts verbieten            */
  oldvect = getvect(SERINT);              /* alten Interruptvektor sichern  */
  setvect(SERINT,servrout);               /* neuen Interruptvektor installieren */
  o8259 = inportb(0x21);                  /* enable Bit beim 8259 setzen    */
  outportb(0x21,o8259 & 0xef);
  outport(LCR+comx,0x80);                 /* Bit 7 LCR setzen -> Baudrate   */
  outport(RBR+comx,BAUDLSB);              /* Baudrate programmieren         */
  outport(IER+comx,BAUDMSB);
  outport(LCR+comx,UEPAR);                /* Parameter programmieren        */
  outportb(IER+comx,0x01);                /* Receive Interrupt am 8250 erlauben */
  outportb(MCR+comx,0x0b);                /* DTR,RTS und OUT2 setzen        */
                                          /* (=Hardware Int. enable)        */
  enable();                               /* Interrupts wieder erlauben     */
```

```
}                                          /* end of install                    */


/************************** DINSTALL ***************************************/
/* Beendet die Kommunikation. setzt Interruptvektoren wieder zurueck          */

void dinstall()
{
  disable();                               /* Interrupts verbieten              */
  outportb(0x21,o8259);                    /* COM Interrupt am 8259 verbieten   */
  outportb(IER+comx,0);                    /* Interrupts am 8250 verbieten      */
  outportb(MCR+comx,0x03);                 /* Leitung out2 auf 0 setzen         */
                                           /* => Interrupt enable abklemmen     */
  setvect(SERINT,oldvect);                 /* Interruptvektor wieder einstellen */
  enable();                                /* Interrupts erlauben               */
}                                          /* end of dinstall                   */


/************************** SENBYT *****************************************/
/* Ein Zeichen ueber serielle Schnittstelle senden                            */

senbyt(CHAR c)
{
  while (! (inportb(LSR+comx) & 0x20)); /* warten bis TX - hold. Reg. leer   */
  outportb(RBR+comx,c);                    /* Zeichen ausgeben                  */
}                                          /* end of senbyt                     */


/************************** WARTE ******************************************/
/* Wartet in einer Schleife, abhaengig vom uebergebenen Parameter p           */

void warte(int p)
{
int q;
  for (q=1;q<p;q++);
}                                          /* end of warte                      */


/************************** BEFEHLE ****************************************/
/* Gibt Befehlsfolge an den MVUS langsam aus (simuliert Tasteneingabe).       */

void befehle(char *sadr)
{
  while (*sadr != 0)                       /* bis Ende Befehlsstring            */
  {
    warte(VERZ);                           /* MVUS Zeit fuer Verarbeitung geben */
    senbyt(toupper(*sadr++));              /* in Grosschrift senden             */
  }
  warte(VERZ/2);
}                                          /* end of befehle                    */


/************************** OERWEIT ****************************************/
```

```
/* Entfernt Extension vom Workfilenamen.                                    */
/* Liefert den String Temp                                                  */

void oerweit(void)
{
  strcpy(wfohne,workfile);
  *strchr(wfohne,'.')=0;
  return;
}                                        /* end of oerweit                  */


/************************** GETWF ****************************************/
/* Liest Workfilename, wenn nicht schon vorhanden                           */
/* Liefert 1 wenn kein Worfilename angegeben wurde                          */

int getwf(void)
{
  if (workfile[0] == 0)                  /* noch kein Workfile da ?         */
  {
    printf("\nWorkfile : ");             /* Name holen                      */
    gets(workfile);
    if (workfile[0]==0)  return(1);      /* Kein Name eingegeben ?          */
    if (strchr(workfile,'.')==NULL) {strcat(workfile,".MAC");}

    oerweit();                           /* Name ohne Extens. erzeugen      */
  }
  return(0);
}                                        /* end of getwf                    */


/************************** ASEDITOR ***************************************/
/* Ruft den Editor auf                                                      */
/* liefert 1 wenn kein Workfilename vorhanden                               */

int aseditor(void)
{
  printf(" EDITOR");
  if (getwf()!=0) return(1);             /* Worfilename holen               */
  spawnlp(P_WAIT,EDIT," ",workfile,NULL);
                                         /* Editor im child-process         */
  return(0);
}                                        /* end of aseditor                 */


/************************** ASENDE ****************************************/
/* Sendet Textfile an den MVUS                                              */
/* Liefert im Fehlerfall 1                                                  */

int asende(void)
{
  empfang='f';                           /* Empfang abschalten              */
  printf("\n");
  adresse=ANFTEX;                        /* Anfang Textbereich MVUS         */
  zeile=0;                               /* Zeilenzaehler                   */
  if((fl=fopen(workfile,"r"))==NULL)     /* Workfile oeffnen                */
```

```
{
  printf("\n\bDie Datei %s ist nicht vorhanden ! ",workfile);
  return(1);
}
pu1 = anfpu1 = recbuf;                  /* Ringpuffer als Zwischenspeicher   */
while (fgets(puffer,PLEN,fl) != NULL) /* zeilenweise lesen                   */
{
  adresse=adresse+ZLEN;                 /* Anfangsadresse der naechsten Zeile */
  zeile++;                              /* Zeilenzaehler erhoehen             */
  anfpu1=pu1;                           /* anfpu1=Anfangsadresse dieser Zeile */
                                        /* im Puffer                          */
  *pu1++=(adresse%256);                 /* Adresse fuer MVUS Editor Verkettung*/
  *pu1++=(adresse/256);                 /* ablegen                            */
  for (i=0;i < (strlen(puffer)-1);i++) /* eingelesene Zeile abtasten          */
  {
    if (puffer[i]==TAB)                 /* Zeichen=Tabulator ?                */
    {
      aktpos=pu1-anfpu1-2;              /* aktpos=aktuelle Position in Zeile  */

      /* Tabulatorweite ist im MVUS Editor nicht einheitlich !               */
      /*.Tab in Leerzeichen wandeln                                          */

      if ((aktpos < TABANZ) || (aktpos >= (3*TABANZ)))
        for (k=0;k<(TABANZ-((aktpos)%TABANZ));k++)
          *pu1++=' ';
      else
        if (aktpos <(2*TABANZ-2))
          for (k=0;k<(TABANZ-2-((aktpos)%TABANZ));k++)
            *pu1++=' ';
        else
          for (k=0;k<(3*TABANZ)-(aktpos);k++)
            *pu1++=' ';
    }
    else                                /* kein TAB. in Grosschrift wandeln   */
      *pu1++=toupper(puffer[i]);
  }
  if ((pu1-anfpu1) < MAXLEN)            /* Maximale Zeilenlaenge              */
  {
    aktpos=pu1-anfpu1;
    for (i=0;i<=(MAXLEN -aktpos);i++) *pu1++=' ';
      *pu1++=CR;                        /* Mit ' ' auffuellen                 */
  }
  else
  {
    pu1=anfpu1+MAXLEN+1;                /* Zu lange Zeilen abschneiden        */
    *pu1++=CR;
  }
}
fclose(fl);                             /* Textdatei schliessen               */
*pu1++=0;                               /* Puffer zu String machen            */
laenge=strlen(recbuf);                  /* Laenge feststellen                 */
*(pu1-ZLEN)  =ANFTEX%256;               /* Verkettung schliessen              */
*(pu1-ZLEN-1)=ANFTEX/256;
sprintf(bef,"I%04x %04x 0a\r",ANFTEX,ANFTEX+laenge);
befehle(bef);                           /* MVUS empfangsbereit machen         */
printf("\nUebertragung von %s !  Bitte warten !\n",workfile);
for(i=0;i<=laenge;senbyt(recbuf[i++])) /* Text uebertragen                    */
pu1=wadr=radr=recbuf;                   /* Puffer initialisieren              */
```

```
  /* Datenbereich des MVUS-Editors patchen                                  */
  /* Siehe Assemblerlisting des MVUS-Editors                                */
  *pu1++=0; *pu1++=0;                  /* ndel Zahl der geloeschten Saetze  */
  *pu1++=zeile%256;                    /* nzei Zahl der Zeilen              */
  *pu1++=zeile/256;
  *pu1++=1; *pu1++=0;                  /* lzei laufende Zeilennummer =1     */
  zeile=ANFTEX+(ZLEN*zeile);           /* zeile=Adresse des laufenden Satzes*/
  *pu1++=(zeile-ZLEN)%256;             /* advor Adresse des Vorgaengers     */
  *pu1++=(zeile-ZLEN)/256;
  *pu1++=zeile%256;                    /* adlau Adresse des laufenden Satzes*/
  *pu1++=zeile/256;
  *pu1++=(zeile+ZLEN)%256;             /* adnaf Adresse des Nachfolgers     */
  *pu1++=(zeile+ZLEN)/256;
  *pu1++=0xff;*pu1++=0xff;             /* delan Adresse erster geloeschter  */
  *pu1++=ANFTEX%256;                   /* adanf Adresse des ersten Satzes   */
  *pu1++=ANFTEX/256;
  *pu1++=ANFTEX%256;                   /* adret Adresse in Kette            */
  *pu1++=ANFTEX/256;
  *pu1++=(zeile+ZLEN)%256;             /* adneu                             */
  *pu1++=(zeile+ZLEN)/256;
  *pu1++=ANFTEX%256;                   /* polau                             */
  *pu1++=ANFTEX/256;
  laenge=22;                           /* Anzahl der zu patchenden Bytes    */
  sprintf(bef,"I%04x %04x 0a\r",ANFDAT,ANFDAT+laenge);
  befehle(bef);
  for(i=0;i<=laenge;senbyt(recbuf[i++]))  /* senden                         */
  wadr=radr=pu1=recbuf;                /* Puffer initialisieren             */
  return(0);                           /* end of asende                     */
}


/***************************** GIVEHELP ***************************************/
/* Gibt bei falschen Befehlsparametern einen Hilfetext aus.                   */

void givehelp (void)
{
  printf("\nAufruf: MVUS [/mx] [/cy] [/lz] [File.ext] \n\n");
  printf("Parameter /m : Modus   x=0..3   default: Modus 2\n");
  printf("          /c : COMy:   y=1..4   default: COM1:\n");
  printf("          /l : LPTz:   z=1..2   default: LPT1:\n\n");
  printf("[] : Parameter ist optional\n");
  printf("Bei Modus 3 muss ein Filename mit Erweiterung angegeben werden.\n");
}                                      /* end of givehelp                   */


/***************************** SENDECOM ***************************************/
/* Sendet Binaerfile zum MVUS                                                 */

int sendecom (char *fl)
{
  struct stat stbuf;
  FILE *fp;
  unsigned int anzahl,laenge,anfang,weiter;
  CHAR befe[30];
  char zeichen;
```

```
  empfang='f';
  if((fp=fopen(fl,"r"))==NULL)          /* Binaeres File oeffnen              */
  {
    printf("%s kann nicht geöffnet werden. \n",fl);
    return(1);
  }
  else
  {
    fstat(fileno(fp),&stbuf);           /* Status des geoeffneten Files holen */
    anfang=fgetc(fp);                   /* erstes Byte lesen                  */
    if (anfang != 0xc3)                 /* ersten drei Byte = JMP xxxx        */
    {
      printf("\nFile nicht übertragbar! ORG muß >= 103h sein!");
      return(1);
    }
    anfang=fgetc(fp);
    anfang=anfang+((fgetc(fp)-1)*256); /* 100h abziehen,da fcb 100h Bytes     */
    printf("Start  : %0xh\n",anfang+256);
    laenge=stbuf.st_size-anfang;        /* Anzahl zu sendenden Bytes holen    */
    printf("Länge  : %xh\n",laenge);
    weiter=anfang-1;
    sprintf(befe,"I%04x %04x 0A\r",(anfang+256),(anfang+255+laenge));
    befehle(befe);                      /* MVUS empfangsbereit machen         */
    fseek(fp,anfang,0);                 /* auf erstes Byte stellen            */
    for(anzahl=0;anzahl < laenge;anzahl++)
    {
      fseek(fp,++weiter,0);             /* auf naechstes Byte stellen         */
      zeichen=fgetc(fp);                /* so umstaendlich, da EOF hier nicht */
      if (feof(fp)) zeichen=0x1a;       /* das Dateiende markiert             */
      senbyt(zeichen);                  /* Byte senden                        */
    }
    fclose(fp);                         /* Binaerdatei schliessen             */
    printf("File wurde übertragen.\n");
  }
  return(0);                            /* alles ok                           */
}                                       /* end of sendecom                    */


/***************************** ATEXTAUS ************************************/
/* Holt Programmtext aus dem Editor des MVUS und schreibt ihn in ein File    */
/* Gibt im Fehlerfall 1 zurueck                                              */

int atextaus(void)
{
  printf("\nAusgabefile : ");           /* Name des Ausgabefiles holen        */
  gets(outfile);
  if(outfile[0]==0) return(1);          /* Name eingegeben ?                  */
  if(strchr(outfile,'.')==NULL) strcat(outfile,".MAC");
                                        /* Erweiterung anhaengen              */
  if((fl=fopen(outfile,"r"))!=NULL)     /* Datei schon vorhanden ?            */
  {
    fclose(fl);
    printf("\nDie Datei %s ist schon vorhanden. Überschreiben ? (j/n)",
            outfile);
    while(!kbhit());                    /* auf Tastendruck warten             */
    if (toupper(bdos(7,0,0))!='J') return(1);
  }
```

```
  if((fl=fopen(outfile,"w"))==NULL)     /* Datei anlegen                    */
  {
    printf("\n\bDie Datei %s kann nicht angelegt werden ! ",outfile);
    return(1);
  }
  wadr=radr=recbuf;                     /* Puffer initialisieren            */
  printf("\nText wird zu %s übertragen. Bitte warten !\n",outfile);
  sprintf(bef,"O%04x %04x 0A\r",ANFDAT+2,ANFDAT+11);
  befehle(bef);                         /* MVUS Editor-Daten holen          */
  while((wadr-recbuf)<32);              /* warten bis Daten im Puffer       */
  zeile=recbuf[24]+256*recbuf[25];      /* Daten aus Puffer auslesen        */
  ende=(recbuf[28]+256*recbuf[29])+ZLEN;
  wadr=radr=recbuf;                     /* Puffer ruecksetzen               */
  warte(VERZ);                          /* Bei AT's muss gewartet werden    */
  sprintf(bef,"O%04x %04x 0A\r",ANFTEX,ende); /* Aus Editor lesen           */
  befehle(bef);
  while ((wadr-recbuf) < (ende+41-ANFTEX)); /* warten bis ganzer Text drin    */
  warte(VERZ);
  adresse=42;                           /* Verkettete Liste aufbereiten     */
  for (i=1;i<=zeile;i++)                /* jeweils 1 Zeile                  */
  {
    for(k=adresse+2;k<=adresse+MAXLEN;k++) /* Zeile byteweise uebertragen     */
    {
      putc(recbuf[k],fl);putchar(recbuf[k]); /* auf File und Screen          */
    }
    putc(LF,fl);putchar(LF);            /* Linefeed anhaengen               */
                                        /* Wo ist die naechste Zeile ?      */
    adresse=(recbuf[adresse]+256*recbuf[adresse+1])-ANFTEX+42;
  }                                     /* alle Zeilen gelesen              */
  fclose(fl);                           /* Ausgabefile schliessen           */
  wadr=radr=recbuf;                     /* Ringpuffer wieder initialisieren */
  return(0);
}                                       /* end of atextaus                  */



/***************************** FUNKHLP ***************************************/
/* Bringt abhaengig vom Modus die Belegung der Funktionstasten auf den       */
/* Bildschirm                                                                */

void funkhlp(void)
{
  printf("\n\n\n\Funktionstasten im Modus %c ",modus);
  switch (modus){
    case '0' : printf("(reiner Terminal-Betrieb) :\n\n");break;
    case '1' : printf("(Betrieb mit der Assemblerkarte des MVUS) :\n\n");break;
    case '2' : printf("(Betrieb mit CPM-Emulator KCE) :\n\n");break;
  }
  printf("    F1 : Bringt diese Hilfe auf den Bildschirm\n");
  if (modus!='0') printf("    F2 : Drucken von Files\n");
  switch (modus){
    case '1' :{
      printf("    F3 : Ruft den Editor auf und überträgt das Workfile\n"
             "         nach Verlassen des Editors automatisch zum MVUS\n");
      printf("    F4 : Liest einen neuen Workfilenamen ein\n");
      printf("    F5 : Ruft den Editor auf ohne das Workfile nach\n"
             "         Verlassen des Editors zum MVUS zu senden\n");
      printf("    F6 : Sendet das Workfile zum MVUS\n");
```

```
      printf("    F7 : Holt ein Programm aus dem Editor des MVUS und\n"
             "         speichert es unter dem angegebenen Namen ab\n");
      printf("    F8 : Nicht belegt\n");
      printf("    F9 : Nicht belegt\n");
      break;
    }
    case '2' :{
      printf("    F3 : Ruft den Editor auf (mit Workfile)\n");
      printf("    F4 : Liest einen neuen Workfilenamen ein\n");
      printf("    F5 : Ruft den M80-Assembler auf (ohne Workfile)\n");
      printf("    F6 : Ruft den L80-Linker auf (ohne Workfile)\n");
      printf("    F7 : Sendet den Programmcode (Workfile) zum MVUS\n");
      printf("    F8 : Ruft nacheinander den Assembler und Linker auf\n"
             "         und sendet den Programmcode (Workfile) zum MVUS\n");
      printf("    F9 : Ruft den Debugger DDT auf (mit Workfile)\n");
      break;
    }
  }
  printf("   F10 : Beendet das Programm\n");
  printf("Alt-F1 : Schaltet einen Modus weiter.\n");
  return;
}                                        /* end of funkhlp                    */


/*************************** DRUCKE ****************************************/
/* Gibt das angegebene File auf dem Drucker aus.                            */
/* Liefert 1, wenn das zu druckende File nicht gefunden wurde.              */

int drucke(void)
{
  FILE *drk,*fdrk;
  unsigned char zeichen[254];

  printf("\nName des zu druckenden Files (mit Erw.) : "); /* Name holen     */
  gets(outfile);
  if (outfile[0]==0)  return(1);        /* Kein Name eingegeben ?            */
  if((fdrk=fopen(outfile,"r"))==NULL){
    printf("\nDie Datei %s kann nicht geöffnet werden !\n",outfile);
    return(1);
  }
  if((drk=fopen(lpt,"w"))==NULL){
    printf("\nDer Drucker ist nicht bereit !\n");
    return(1);
  }
  printf("\nMit 132 Zeichen/Zeile drucken? (j/n)");
  while(!kbhit());                      /* auf Tastendruck warten            */
  if (toupper(bdos(7,0,0))=='J')
    fputs(DRK132,drk);
  else
    fputs(DRK80,drk);
  printf("\n\n Abbruch mit beliebiger Taste\n");
  while((fgets(zeichen,250,fdrk))!=NULL){
    if (kbhit()) {bdos(7,0,0);break;}
    if ((fputs(zeichen,drk))==EOF){
      printf("Fehler bei Ausgabe an den Drucker\n");
      break;
    }
```

```
  }
  fputs(DRK80,drk);
  fclose(drk);
  fclose(fdrk);
  return(0);
}                                       /* end of drucke                    */


/*****************************************************************************/
/*                          Hauptprogramm                                    */
/*****************************************************************************/

int cdecl main(int argc, char *argv[])
{
  extern char *optarg;                  /* include-Datei getopt              */
  extern int opterr,optind;
  extern int getopt(int argc,char *argv[],char *optionS);
  funct      ftaste;

  modus='2';                            /* default-Werte einstellen          */
  com='1';
  sprintf(lpt,"LPT1");
  opterr=FALSE;                         /* eigene Fehlerbehandlung           */
  while((opt=getopt(argc,argv,"M:C:m:c:L:l:")) != EOF)  /* Parameter holen  */
  {
    switch (opt)
    {
      case 'M' :
      case 'm' : { if ((*optarg>='0')&&(*optarg<='3')) {
                     modus=*optarg; break;
                   }
                   else {
                     printf("Falscher Parameter\n");
                     givehelp();
                     return(1);         /* errorlevel setzen                 */
                   }
                 }
      case 'C' :
      case 'c' : { if ((*optarg>='1')&&(*optarg<='3')) {
                     com=*optarg; break;
                   }
                   else {
                     printf("Falscher Parameter\n");
                     givehelp();
                     return(1);         /* errorlevel setzen                 */
                   }
                 }
      case 'L' :
      case 'l' : { if ((*optarg>='1')&&(*optarg<='2')) {
                     if (*optarg=='2') sprintf(lpt,"LPT2");
                     break;
                   }
                 }
      case '?' : { printf("Falscher Parameter\n");
                   givehelp();
                   return(1);           /* errorlevel setzen                 */
                 }
```

```
  }
}
if (modus=='3')                          /* nur com-File senden              */
{
  ifile=argv[optind];                    /* File-Namen holen                 */
  if (*ifile==' ' || optind>=argc)       /* gueltig ?                        */
  {
     printf("Bei Modus 3 muss der Name (mit Extension) \n"
            "des zu sendenden Files angegeben werden.");
    return(1);                           /* errorlevel setzen                */
  }
}

switch (com){
  case '2' : {comx=0x2f0; break;}        /* Basisadressen fuer COM           */
  case '3' : {comx=0x3e0; break;}
  case '4' : {comx=0x2e0; break;}
  default  : {comx=0x3f0;}
}
empfang='f';                             /* Empfang abschalten               */
printf("%s",CLRSCR);                     /* Bildschirm loeschen              */
printf("\n\t\t    MVUS-IBM/AT - 26.1.88\n\n");

workfile[0] = 0;                         /* Workfilenamen loeschen           */
wadr = radr = recbuf;                    /* Ringpuffer initialisieren        */
install();                               /* Interrupt umleiten               */
if (modus=='3')                          /* nur senden ?                     */
{
  ret=sendecom(ifile);                   /* Binaerfile senden                */
  befehle("G1\r");                       /* MVUS zuruecksetzen               */
  dinstall();                            /* Int-Vektor wiederherstellen      */
  return(ret);                           /* zu DOS zurueck                   */
}
printf("\nSie sind im Modus %c ",modus);
switch (modus){
  case '0' : printf("(reiner Terminal-Betrieb) :\n\n");break;
  case '1' : printf("(Betrieb mit der Assemblerkarte des MVUS) :\n\n");break;
  case '2' : printf("(Betrieb mit CPM-Emulator KCE) :\n\n");break;
}
befehle("G1\r");                         /* Reset fuer MVUS                  */
while (1)                                /* raus durch break                 */
{
  empfang='t';                           /* Empfang einschalten              */
  if (kbhit())                           /* Liegt Zeichen von Tastatur an?   */
  {
    ckb = toupper(bdos(0x07,0,0));       /* Zeichen von Tastatur liegt an    */
    if (ckb==0)                          /* Funktionstaste  ?                */
    {
      ckb=bdos(0x07,0,0);                /* ja->Tastatur erneut lesen        */
      ftaste=(ckb-59);
      switch (ftaste){
        case f10 : {
          dinstall(); return(0);         /* F10 = beendet                    */
        }
        case AF1 : {                     /* Alt-F1 = Modus=Modus+1           */
          modus++;
          if (modus >= '3') modus='0';
          printf("\nSie sind nun im Modus %c. ",modus);
```

```
      switch (modus){
        case '0' : printf("(reiner Terminal-Betrieb) :\n\n");break;
        case '1' : printf("(Betrieb mit der Assemblerkarte des MVUS)"
                          " :\n\n");break;
        case '2' : printf("(Betrieb mit CPM-Emulator KCE) :\n\n");break;
      }
      break;
    }
    case f1 : {
      funkhlp();
      break;
    }
    case f2 : {
      if (modus!='0'){
        drucke();
        break;
      }
    }
  }
  switch (modus){
    case '1' :{                     /* Terminal + MVUS Assemblerkarte     */
      switch (ftaste){
        case f5 :                   /* F5 nur Editor aufrufen             */
        case f3 :{                  /* F3 Editor aufrufen, Text senden    */
          if (aseditor()!=0) break;
          if (ftaste==f5) break;
        }
        case f6 :{                  /* F6 Worfiletext direkt senden       */
          if (getwf()!=0) break;
          asende();
          break;
        }
        case f4 :{                  /* F4 Neuer Workfilename              */
          printf("\nWorkfile : ");
          gets(workfile);           /* Name holen                         */
          if (strchr(workfile,'.')==NULL) strcat(workfile,".MAC");
          oerweit();                /* Workfilename ohne Extens. erzeugen */
          break;
        }
        case f7 :{                  /* Text aus dem MVUS-Editor holen     */
          atextaus();
          break;
        }
      }                             /* switch (ftaste)                    */
      break;
    }                               /* case '1'                           */
    case '2' :{
      switch (ftaste){
        case f3 :{                  /* Nur Editor rufen                   */
          aseditor();
          break;
        }
        case f4 :{                  /* F4 Neuer Workfilename              */
          printf("\nWorkfile : ");
          gets(workfile);           /* Name holen                         */
          if (strchr(workfile,'.')==NULL) strcat(workfile,".MAC");
          oerweit();                /* Workilename ohne Extens. erzeugen  */
          break;
```

```
            }
            case f8 :
            case f5 :{
              if (getwf()!=0) break;
              strcpy(outfile,"M80 ");
              if (ftaste==f8){
                strcat(outfile,wfohne);
                strcat(outfile,",");
                strcat(outfile,wfohne);
                strcat(outfile,LST);
                strcat(outfile,wfohne);
                strcat(outfile,"/L/R");
              }
              spawnlp(P_WAIT,"kce"," ",outfile,NULL);
              if (ftaste!=f8) break;
            }
            case f6 :{
              if (getwf()!=0) break;
              strcpy(outfile,"L80 ");
              if (ftaste==f8){
                strcat(outfile,wfohne);
                strcat(outfile,",");
                strcat(outfile,wfohne);
                strcat(outfile,"/N/E");
              }
              spawnlp(P_WAIT,"kce"," ",outfile,NULL);
              if (ftaste!=f8) break;
            }
            case f7 :{
              if (getwf()!=0) break;
              strcpy(outfile,wfohne);
              strcat(outfile,".com");
              sendecom(outfile);
              break;
            }
            case f9 :{
              if (getwf()!=0) break;
              strcpy(outfile,"DDT ");
              strcat(outfile,wfohne);
              strcat(outfile,".com");
              spawnlp(P_WAIT,"kce"," ",outfile,NULL);
              break;
            }
          }
          break;
        }                              /* case '2'                          */
      }                                /* switch (modus)                    */
      empfang='f';
      befehle("G1\r");
      empfang='t';
    }                                  /* if ckb==0                         */
    else senbyt(ckb);
  }                                    /* if kbhit                          */
  while (wadr != radr)
  {
    disable();                         /* Interrupt ausschalten             */
    c = *radr++;                       /* Zeichen aus Ringpuffer holen      */
    if (radr>&recbuf[sizeof(recbuf)-1]) radr=recbuf; /* Lesepointer bei     */
```

```
                                        /* Ueberlauf zuruecksetzen          */
        enable();                       /* Interrupt wieder erlauben        */
        if (c == 12) c=0x20;            /* empfangenes Zeichen anzeigen     */
        if (c == 127) c='.';            /* Drucker hat Probleme mit 127     */
        if (c != 0x1a) putchar(c & 127);
      }                                 /* while wadr != radr               */
    }                                   /* while 1                          */
}                                       /* end of main                      */
```

```
                0190 >; AUSGABETEXTE
                0191 >ET1 DC 'FEHLER: SPEICHERSCHUTZ ?'
2 46 45 48 4C 45 52 3A 20 53 50 45 49 43 48 45 52 53 43 48 55 54 5A 20 BF
                0192 >ET2 DC 'EDITOR 1.0 Alt/Neu? A/N>'
A 45 44 49 54 4F 52 20 31 2E 30 20 41 6C 74 2F 4E 65 75 3F 20 41 2F 4E BE
                0193 >ET3 DC 'EINGABE'
2 45 49 4E 47 41 42 C5
                0194 >ET4 DC 'FEHLER: RAM VOLL ?'
9 46 45 48 4C 45 52 3A 20 52 41 4D 20 56 4F 4C 4C 20 BF
                0195 >ET5 DC 'KOMMANDO'
8 4B 4F 4D 4D 41 4E 44 CF
                0196 >ET6 DC 'KOMM>'
3 48 4F 4D 4D BE
                0197 >ET7 DC 'FEHLER: KOMMANDO ?'
8 46 45 48 4C 45 52 3A 20 4B 4F 4D 4D 41 4E 44 4F 20 BF
                0198 >ET8 DC ' #NEU>'
A 20 23 4E 45 55 BE
                0199 >ET9 DC '*** ENDE DER ZEILEN ***'
0 2A 2A 20 45 4E 44 45 20 44 45 52 20 5A 45 49 4C 45 4E 20 2A 2A AA
                019A >; ARBEITSSPEICHER
000             019B >          ORG   4000H       ; EDITOR - ASSMEBLER - ARBEITSRA
000             019C >PUFF      DS    60          ; ARBEITSPUFFER
03C             019D >NDEL      DS    2           ; ZAHL DER GELOESCHTEN SAETZE
03E             019E >NZEI      DS    2           ; ZAHL DER ZEILEN EINSCHL. LAUF
040             019F >LZEI      DS    2           ; LAUFENDE ZEILENNUMMER
042             01A0 >ADVOR     DS    2           ; ADRESSE DES VORGAENGERS
044             01A1 >ADLAU     DS    2           ; ADRESSE DES LAUFENDEN SATZES
046             01A2 >ADNAF     DS    2           ; ADRESSE DES NACHFOLGERS
048             01A3 >DELAN     DS    2           ; ADRESSE DES 1. GELOESCHTEN SAT
04A             01A4 >ADANF     DS    2           ; ADRESSE DES 1. SATZES
04C             01A5 >ADRET     DS    2           ; ADRESSE DES NAECHSTEN IN KETTE
04E             01A6 >ADNEU     DS    2           ; ADRESS DES UEBERNAECHSTEN
050             01A7 >POLAU     DS    2           ; LAUFENDE ZEIGERPOSITION
052             01A8 >PONAF     DS    2           ; NACHFOLGENDE ZEIGERPOSITION
054             01A9 >EDIRAM    EQU   4400H.      ; EDITOR - RAM - BEREICH
000             01AA >.         END
```

Anhang, Bildschirminhalt bei Ben. v. MVU nach F1 ↵

Funktionstasten im Modus 2 (Betrieb mit CPM-Emulator KCE) :

F1 : Bringt diese Hilfe auf den Bildschirm
F2 : Drucken von Files
F3 : Ruft den Editor auf (mit Workfile)
F4 : Liest einen neuen Workfilenamen ein
F5 : Ruft den M80-Assembler auf (ohne Workfile)
F6 : Ruft den L80-Linker auf (ohne Workfile)
F7 : Sendet den Programmcode (Workfile) zum MVUS
F9 : Ruft nacheinander den Assembler und Linker auf
und sendet den Programmcode (Workfile) zum MVUS
F9 : Ruft den Debugger DDT auf (mit Workfile)
F10 : Beendet das Programm
Alt-F1 : Schaltet einen Modus weiter.

Sie sind nun im Modus 0. (reiner Terminal-Betrieb) :

Sie sind nun im Modus 1. (Betrieb mit der Assemblerkarte des MVUS) :

Sie sind nun im Modus 2. (Betrieb mit CPM-Emulator KCE) :

SKELMAC.MAC

```
;SKELETT ZUR ERSTELLUNG VON PROGRAMMEN FUER DEN MVUS-8085.
        ASEG
        ORG     1000H       ;STARTADRESSE FUER MVUS-PROGRAMME
;EINFUEGEN DER BEFEHLSFOLGE AN DIESER STELLE:
START:  MVI     A,20BH      ;INITIALISIERE 8255
        OUT     03H
EIN:    IN      01H         ;EINGABE VON KIPPSCHALTERN
        OUT     00H         ;AUSGABE AUF DIE DIODENANZEIGE
        OUT     18H         ;AUSGABE AUF DIE 7-SEGMENT ANZEIGE
        ORI     00H         ;PRUEFE OB 00 EINGEGEBEN?
        JNZ     EIN         ;WENN NEIN, WEITER MIT EINGABE
ENDE:   RST     20H         ;ZURUECK ZUM BETRIEBSSYSTEM
;EINFUEGEN DER DATEN AN DIESER STELLE
;ORG KANN NEU GESETZT WERDEN, MUSS ABER > ALS DIE OBEN
;FESTGELEGTE ADRESSE SEIN!

        END     START       ;ENDE DES ASSEMBLERPROGRAMMS
```

START OF TEXT

```
;SKELETT ZUR ERSTELLUNG VON PROGRAMMEN FUER DEN MVUS-2085.
        ASEG
        ORG     1000H       ;STARTADRESSE FUER MVUS-PROGRAMME
;EINFUEGEN DER BEFEHLSFOLGE AN DIESER STELLE:
START:  MVI     A,08BH      ;INITIALISIERE 8255
        OUT     03H
EIN:    IN      01H         ;EINGABE VON KIPPSCHALTERN
        OUT     00H         ;AUSGABE AUF DIE DIODENANZEIGE
        OUT     18H         ;AUSGABE AUF DIE 7-SEGMENT ANZEIGE
        ORI     00H         ;PRUEFE OB 00 EINGEGEBEN?
        JNZ     EIN         ;WENN NEIN, WEITER MIT EINGABE
ENDE:   RST     00H         ;ZURUECK ZUM BETRIEBSSYSTEM
;EINFUEGEN DER DATEN AN DIESER STELLE
;ORG KANN NEU GESETZT WERDEN, MUSS ABER > ALS DIE OBEN
;FESTGELEGTE ADRESSE SEIN!


        END     START       ;ENDE DES ASSEMBLERPROGRAMMS
```

END OF TEXT

Funktionstasten im Modus 2 (Betrieb mit CPM-Emulator KCE) :

```
    F1 : Bringt diese Hilfe auf den Bildschirm
    F2 : Drucken von Files
    F3 : Ruft den Editor auf (mit Workfile)
    F4 : Liest einen neuen Workfilenamen ein
    F5 : Ruft den M80-Assembler auf (ohne Workfile)
    F6 : Ruft den L80-Linker auf (ohne Workfile)
    F7 : Sendet den Programmcode (Workfile) zum MVUS
    F8 : Ruft nacheinander den Assembler und Linker auf
         und sendet den Programmcode (Workfile) zum MVUS
    F9 : Ruft den Debugger DDT auf (mit Workfile)
   F10 : Beendet das Programm
Alt-F1 : Schaltet einen Modus weiter.
```

```
;SKELETT ZUR ERSTELLUNG VON PROGRAMMEN FUER DEN MVUS-8085.
        ASEG
        ORG     1000H   ;STARTADRESSE FUER MVUS-PROGRAMME
;EINFUEGEN DER BEFEHLSFOLGE AN DIESER STELLE:
START:  MVI     A,080H  ;INITIALISIERE 8255
        OUT     03H
EIN:    IN      01H     ;EINGABE VON KIPPSCHALTERN
        OUT     20H     ;AUSGABE AUF DIE DIODENANZEIGE
        OUT     19H     ;AUSGABE AUF DIE 7-SEGMENT ANZEIGE
        ORI     00H     ;PRUEFE OB 00 EINGEGEBEN?
        JNZ     EIN     ;WENN NEIN, WEITER MIT EINGABE
ENDE:   RST     00H     ;ZURUECK ZUM BETRIEBSSYSTEM
;EINFUEGEN DER DATEN AN DIESER STELLE
;ORG KANN NEU GESETZT WERDEN, MUSS ABER > ALS DIE OBEN
;FESTGELEGTE ADRESSE SEIN!


        END     START   ;ENDE DES ASSEMBLERPROGRAMMS
```